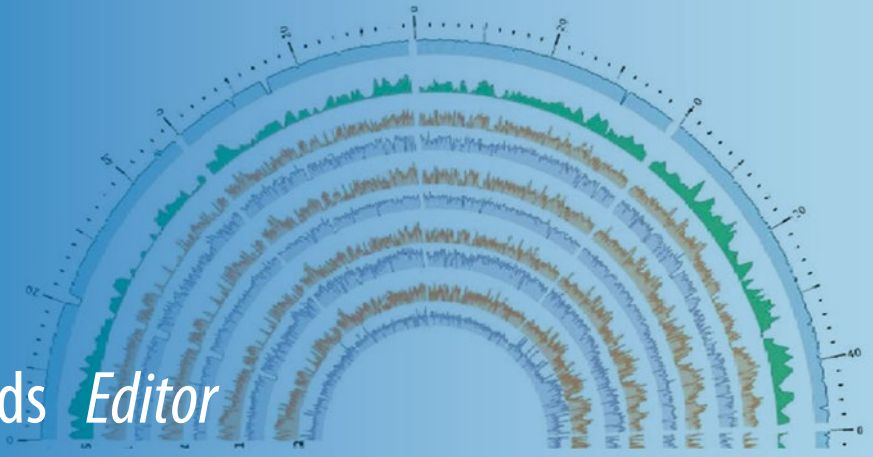


Methods in
Molecular Biology 1374

Springer Protocols

David Edwards *Editor*



Plant Bioinformatics

Methods and Protocols

Second Edition

 Humana Press

METHODS IN MOLECULAR BIOLOGY

Series Editor

John M. Walker

School of Life and Medical Sciences

University of Hertfordshire

Hatfield, Hertfordshire, AL10 9AB, UK

For further volumes:

<http://www.springer.com/series/7651>

Plant Bioinformatics

Methods and Protocols

Second Edition

Edited by

David Edwards

University of Western Australia, School of Plant Biology, Perth, Australia

 **Humana Press**

Editor

David Edwards
University of Western Australia
School of Plant Biology
Perth, Australia

ISSN 1064-3745 ISSN 1940-6029 (electronic)
Methods in Molecular Biology
ISBN 978-1-4939-3166-8 ISBN 978-1-4939-3167-5 (eBook)
DOI 10.1007/978-1-4939-3167-5

Library of Congress Control Number: 2015951438

Springer New York Heidelberg Dordrecht London
© Springer Science+Business Media New York 2016

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

Humana Press is a brand of Springer
Springer Science+Business Media LLC New York is part of Springer Science+Business Media (www.springer.com)

Preface

Bioinformatics is a rapidly growing field of research that is being driven by the requirement to manage and interrogate the vast quantities of data being generated by 'omics technologies. The term bioinformatics means different things to different people, and following the theme of this series, this volume focuses on applied bioinformatics with specific applications to crops and model plants.

This volume is aimed at plant biologists who have an interest in, or requirement for, accessing and manipulation of the huge amount of data being generated by high-throughput technologies. The volume would also be of interest to bioinformaticians and computer scientists who would benefit from an introduction to the different tools and systems available for plant research.

The scope of bioinformatics now extends from the genome to the phenome and is increasingly being applied outside of pure research and towards supporting the accelerated breeding of crop plants.

It is the integration of information relating to heritable agronomic traits, including important metabolic profiles, with the emerging genome and transcriptome data that will drive plant research, crop breeding, and bioinformatics developments in the future. One observation during the production of this volume is the requirement to manage the increasing volume and diversity of data from different plants and also the integration of multiple diverse forms of data. I expect this trend to continue as this field of research continues to develop.

Scientific research by its nature progresses and changes and this is especially true for bioinformatics. The rapid evolution of the tools and systems described in this volume may change during the lifetime of this edition and it is suggested that the reader consult the relevant web pages directly as they frequently host detailed list of updates and changes.

Perth, WA, Australia

David Edwards

Contents

<i>Preface</i>	<i>v</i>
<i>Contributors</i>	<i>ix</i>
1 Using GenBank <i>Eric W. Sayers and Ilene Karsch-Mizrachi</i>	1
2 UniProtKB/Swiss-Prot, the Manually Annotated Section of the UniProt KnowledgeBase: How to Use the Entry View <i>Emmanuel Boutet, Damien Lieberherr, Michael Tognolli, Michel Schneider, Parit Bansal, Alan J. Bridge, Sylvain Poux, Lydie Bougueleret, and Ioannis Xenarios</i>	23
3 KEGG Bioinformatics Resource for Plant Genomics and Metabolomics. <i>Minoru Kanehisa</i>	55
4 Plant Pathway Databases <i>Pankaj Jaiswal and Björn Usadel</i>	71
5 The Plant Ontology: A Tool for Plant Genomics <i>Laurel Cooper and Pankaj Jaiswal</i>	89
6 Ensembl Plants: Integrating Tools for Visualizing, Mining, and Analyzing Plant Genomics Data <i>Dan Bolser, Daniel M. Staines, Emily Pritchard, and Paul Kersey</i>	115
7 Gramene: A Resource for Comparative Analysis of Plants Genomes and Pathways. <i>Marcela Karey Tello-Ruiz, Joshua Stein, Sharon Wei, Ken Youens-Clark, Pankaj Jaiswal, and Doreen Ware</i>	141
8 PGSB/MIPS Plant Genome Information Resources and Concepts for the Analysis of Complex Grass Genomes <i>Manuel Spannagl, Kai Bader, Matthias Pfeifer, Thomas Nussbaumer, and Klaus F.X. Mayer</i>	165
9 MaizeGDB: The Maize Genetics and Genomics Database <i>Lisa Harper, Jack Gardiner, Carson Andorf, and Carolyn J. Lawrence</i>	187
10 WheatGenome.info: A Resource for Wheat Genomics Resource <i>Kaitao Lai</i>	203
11 User Guidelines for the Brassica Database: BRAD <i>Xiaobo Wang, Feng Cheng, and Xiaowu Wang</i>	215
12 TAG Sequence Identification of Genomic Regions Using TAGdb <i>Pradeep Ruperao</i>	233
13 Short Read Alignment Using SOAP2 <i>Bhavna Hurgobin</i>	241

14 Tablet: Visualizing Next-Generation Sequence Assemblies
and Mappings 253
*Iain Milne, Micha Bayer, Gordon Stephen, Linda Cardle,
and David Marshall*

15 Analysis of Genotyping-by-Sequencing (GBS) Data. 269
*Sateesh Kagale, Chushin Koh, Wayne E. Clarke, Venkatesh Bollina,
Isobel A.P. Parkin, and Andrew G. Sharpe*

16 Skim-Based Genotyping by Sequencing Using a Double
Haploid Population to Call SNPs, Infer Gene Conversions,
and Improve Genome Assemblies 285
Philipp Emanuel Bayer

17 Finding and Characterizing Repeats in Plant Genomes 293
Jacques Nicolas, Pierre Peterlongo, and Sébastien Tempel

18 Analysis of RNA-Seq Data Using TopHat and Cufflinks 339
Sreya Ghosh and Chon-Kit Kenneth Chan

Index. 363

Contributors

- CARSON ANDORF • *Maize Genetics and Genomics Database, USDA-ARS, Corn Insects and Crop Genetics Research Unit, Iowa State University, Ames, IA, USA*
- KAI BADER • *Plant Genome and Systems Biology, Helmholtz Center Munich, Neuherberg, Germany*
- PARIT BANSAL • *Swiss Institute of Bioinformatics, Centre Medical Universitaire, Geneva, Switzerland*
- MICHA BAYER • *Information & Computational Sciences, The James Hutton Institute, Invergowrie, Dundee, Scotland, UK*
- PHILIPP EMANUEL BAYER • *School of Plant Biology, University of Western Australia, Perth, WA, Australia*
- VENKATESH BOLLINA • *Agriculture and Agri-Food Canada, Saskatoon, SK, Canada*
- DAN BOLSER • *European Molecular Biology Laboratory, European Bioinformatics Institute, Hinxton, Cambridge, UK*
- LYDIE BOUGUELERET • *Swiss Institute of Bioinformatics, Centre Medical Universitaire, Geneva, Switzerland*
- EMMANUEL BOUTET • *Swiss Institute of Bioinformatics, Centre Medical Universitaire, Geneva, Switzerland*
- ALAN J. BRIDGE • *Swiss Institute of Bioinformatics, Centre Medical Universitaire, Geneva, Switzerland*
- LINDA CARDLE • *Information & Computational Sciences, The James Hutton Institute, Invergowrie, Dundee, Scotland, UK*
- CHON-KIT KENNETH CHAN • *School of Plant Biology, University of Western Australia, Crawley, Perth, WA, Australia*
- FENG CHENG • *Institute of Vegetables and Flowers, Chinese Academy of Agricultural Sciences, Beijing, China*
- WAYNE E. CLARKE • *Agriculture and Agri-Food Canada, Saskatoon, SK, Canada*
- LAUREL COOPER • *Department of Botany and Plant Pathology, Oregon State University, Corvallis, OR, USA*
- JACK GARDINER • *Maize Genetics and Genomics Database, USDA-ARS, Corn Insects and Crop Genetics Research Unit, Iowa State University, Ames, IA, USA*
- SREYA GHOSH • *Department of Biotechnology, National Institute of Technology, Durgapur, West Bengal, India*
- LISA HARPER • *Maize Genetics and Genomics Database, USDA-ARS, Corn Insects and Crop Genetics Research Unit, Iowa State University, Ames, IA, USA*
- BHAVNA HURGOBIN • *University of Queensland, St. Lucia, QLD, Australia; School of Plant Biology, University of Western Australia, Perth, WA, Australia*
- PANKAJ JAISWAL • *Department of Botany and Plant Pathology, Oregon State University, Corvallis, OR, USA*
- SATEESH KAGALE • *National Research Council Canada, Saskatoon, SK, Canada*
- MINORU KANEHISA • *Institute for Chemical Research, Kyoto University, Kyoto, Japan*

- ILENE KARSCH-MIZRACHI • *National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Bethesda, MD, USA*
- PAUL KERSEY • *European Molecular Biology Laboratory, European Bioinformatics Institute, Hinxton, Cambridge, UK*
- CHUSHIN KOH • *National Research Council Canada, Saskatoon, SK, Canada*
- KAITAO LAI • *School of Agriculture and Food Science, University of Queensland, St. Lucia, QLD, Australia*
- CAROLYN J. LAWRENCE • *Department of Genetics, Development and Cell Biology, Roy J Carver Co-Laboratory, Iowa State University, Ames, IA, USA*
- DAMIEN LIEBERHERR • *Swiss Institute of Bioinformatics, Centre Medical Universitaire, Geneva, Switzerland*
- DAVID MARSHALL • *Information & Computational Sciences, The James Hutton Institute, Invergowrie, Dundee, Scotland, UK*
- KLAUS F.X. MAYER • *Plant Genome and Systems Biology, Helmholtz Center Munich, Neuherberg, Germany; School of Life Sciences Weihenstephan, Technical University Munich, Neuherberg, Germany*
- IAIN MILNE • *Information & Computational Sciences, The James Hutton Institute, Invergowrie, Dundee, Scotland, UK*
- MARCELA KAREY TELLO-RUIZ • *Cold Spring Harbor Laboratory, Cold Spring Harbor, NY, USA*
- JACQUES NICOLAS • *Dyliss Team, Irisa/Inria Centre de Rennes Bretagne Atlantique, Rennes Cedex, France*
- THOMAS NUSSBAUMER • *Plant Genome and Systems Biology, Helmholtz Center Munich, Neuherberg, Germany*
- ISOBEL A.P. PARKIN • *Agriculture and Agri-Food Canada, Saskatoon, SK, Canada*
- PIERRE PETERLONGO • *Irisa/Inria Centre de Rennes Bretagne Atlantique, Rennes Cedex, France*
- MATTHIAS PFEIFER • *Plant Genome and Systems Biology, Helmholtz Center Munich, Neuherberg, Germany*
- SYLVAIN POUX • *Swiss Institute of Bioinformatics, Centre Medical Universitaire, Geneva, Switzerland*
- EMILY PRITCHARD • *European Molecular Biology Laboratory, European Bioinformatics Institute, Hinxton, Cambridge, UK*
- PRADEEP RUPERAO • *School of Agriculture and Food Sciences, University of Queensland, St. Lucia, QLD, Australia; School of Plant Biology, University of Western Australia, Perth, WA, Australia*
- ERIC W. SAYERS • *National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Bethesda, MD, USA*
- MICHEL SCHNEIDER • *Swiss Institute of Bioinformatics, Centre Medical Universitaire, Geneva, Switzerland*
- ANDREW G. SHARPE • *National Research Council Canada, Saskatoon, SK, Canada*
- MANUEL SPANNAGL • *Plant Genome and Systems Biology, Helmholtz Center Munich, Neuherberg, Germany*
- DANIEL M. STAINES • *European Molecular Biology Laboratory, European Bioinformatics Institute, Hinxton, Cambridge, UK*
- JOSHUA STEIN • *Cold Spring Harbor Laboratory, Cold Spring Harbor, NY, USA*
- GORDON STEPHEN • *Information & Computational Sciences, The James Hutton Institute, Invergowrie, Dundee, Scotland, UK*

- SÉBASTIEN TEMPEL • *LCB, CNRS UMR 7283, Marseille Cedex, France*
- MICHAEL TOGNOLLI • *Swiss Institute of Bioinformatics, Centre Medical Universitaire, Geneva, Switzerland*
- BJÖRN USADEL • *IBMG-Institute for Biology I, RWTH Aachen University, Aachen, Germany; Forschungszentrum Jülich IBG-2 Plant Sciences, Jülich, Germany*
- XIAOBO WANG • *Institute of Vegetables and Flowers, Chinese Academy of Agricultural Sciences, Beijing, China*
- XIAOWU WANG • *Institute of Vegetables and Flowers, Chinese Academy of Agricultural Sciences, Beijing, China*
- DOREEN WARE • *Cold Spring Harbor Laboratory, Cold Spring Harbor, NY, USA; USDA-ARS NAA Plant, Soil & Nutrition Laboratory Research Unit, Cornell University, Ithaca, NY, USA*
- SHARON WEI • *Cold Spring Harbor Laboratory, Cold Spring Harbor, NY, USA*
- IOANNIS XENARIOS • *Swiss Institute of Bioinformatics, Centre Medical Universitaire, Geneva, Switzerland; University of Lausanne, Lausanne, Switzerland*
- KEN YOUENS-CLARK • *Cold Spring Harbor Laboratory, Cold Spring Harbor, NY, USA*

Chapter 1

Using GenBank

Eric W. Sayers and Ilene Karsch-Mizrachi

Abstract

GenBank® is a comprehensive database of publicly available DNA sequences for 300,000 named organisms, more than 110,000 within the embryophyta, obtained through submissions from individual laboratories and batch submissions from large-scale sequencing projects. Daily data exchange with the European Nucleotide Archive (ENA) in Europe and the DNA Data Bank of Japan ensures worldwide coverage. GenBank is accessible through the NCBI Entrez retrieval system that integrates data from the major DNA and protein sequence databases with taxonomy, genome, mapping, protein structure and domain information, as well as the biomedical journal literature in PubMed. BLAST provides sequence similarity searches of GenBank and other sequence databases. Complete bimonthly releases and daily updates of the GenBank database are available by FTP. GenBank usage scenarios ranging from local analyses of the data available via FTP to online analyses supported by the NCBI web-based tools are discussed. To access GenBank and its related retrieval and analysis services, go to the NCBI home page at www.ncbi.nlm.nih.gov.

Key words NCBI, Entrez, DNA Sequence, BLAST, MegaBLAST

1 Introduction

This chapter is designed to serve as a practical guide to using the GenBank nucleotide sequence database in biological research. The chapter is divided into five sections including this introduction. Subheading 2 provides a summary of the content of the database and describes various methods of access. Subheading 3 presents several common ‘methods’ that can be applied to the GenBank data. Subheading 4 provides additional details and examples regarding the described methods. Subheading 5 provides email addresses to use when submitting data to GenBank and to getting help on using the data.

2 Materials

The sections below describe the GenBank database along with its release formats, release cycle, composition, methods of access and integration with other biological resources.

2.1 *The GenBank Database*

GenBank [1] is a comprehensive public database of nucleotide sequences and supporting bibliographic and biological annotation. GenBank is maintained and distributed by the National Center for Biotechnology Information (NCBI), a division of the National Library of Medicine (NLM) at the US National Institutes of Health (NIH) in Bethesda, MD. NCBI builds GenBank from several sources including the submission of sequence data from authors and from the bulk submission of expressed sequence tag (EST), genome survey sequence (GSS), whole genome shotgun (WGS) and other high-throughput data from sequencing centers. The US Office of Patents and Trademarks also contributes sequences from issued patents. GenBank, the European Nucleotide Archive (ENA) [2], and the DNA Databank of Japan (DDBJ) [3] comprise the International Nucleotide Sequence Database Collaboration (INSDC) (www.insdc.org) whose members exchange data daily to ensure a uniform and comprehensive collection of sequence information.

2.1.1 *The GenBank Release Formats and Release Cycle*

NCBI provides free access to GenBank using either FTP or the web-based Entrez search and retrieval system [4]. The FTP release consists of a mixture of compressed and uncompressed ASCII text files, 2052 in release 202, containing sequence data and indices that cross reference author names, journal citations, gene names and keywords to individual GenBank records. For convenience, the GenBank records are partitioned into 19 divisions (*see Note 1*) according to source organism or type of sequence. Records within the same division are packaged as a set of numbered files so that records from a single division may be contained in a series of many files; for example, there are 70 files in the **PLN** division (containing plant and fungal sequences) in release 202. The full GenBank release is offered in two formats; the GenBank ‘flatfile’ format (*see Note 2*), and the more structured and compact Abstract Syntax Notation One (ASN.1) format used by NCBI for internal maintenance. Full releases of GenBank are made every 2 months beginning in the middle of February each year. Between full releases, daily updates are provided on the NCBI FTP site (<ftp.ncbi.nlm.nih.gov/genbank/>, <ftp.ncbi.nlm.nih.gov/ncbi-asn1/>). The Entrez system always provides access to the latest version of GenBank including the daily updates.

2.1.2 The Composition of GenBank

From its inception, GenBank has doubled in size about every 18 months. Release 202, in June 2014, contained 162 billion nucleotide bases from more than 173 million individual sequences. Contributions from WGS projects supplement the data in the traditional divisions to bring the total beyond 780 gigabases. The number of eukaryote genomes for which coverage and assembly are significant continues to increase as well, with 1200 such assemblies now available. Database sequences are classified by and can be queried using a comprehensive sequence-based taxonomy [5] developed by NCBI in collaboration with ENA and DDBJ with the assistance of external advisers and curators. Some 300,000 named species are now represented in GenBank and new species are being added at the rate of over 3000 per month. Detailed statistics for the current release may always be found in the GenBank release notes (<ftp.ncbi.nlm.nih.gov/genbank/gbrel.txt>).

2.1.3 Sources of Plant Sequences

In recent years high-throughput sequencing techniques, such as whole genome shotgun (WGS) sequencing, have become the dominant source of sequence data for many organisms, including green plants. More than 98 % of the sequence data for plants in GenBank release 202 (224 Gbp from some 120,000 plant species) were derived from a high-throughput method, leaving less than 2 % from the traditional PLN division of GenBank. About 80 % of the plant data were produced from WGS methods (Table 1).

Table 1
Distribution of plant sequences among the GenBank divisions in GenBank release 202

Division	Mbp	Fraction (%)
WGS	115,970.4	51.67
CON	66,170.1	29.48
EST	13,441.1	5.99
GSS	11,386.9	5.07
TSA	7186.8	3.20
HTG	6241.7	2.78
PLN	3324.0	1.48
PAT	524.7	0.23
HTC	97.6	0.04
STS	71.7	0.03
ENV	7.9	0.00
SYN	0.7	0.00
Total	224,423.7	100.00

Table 2
Prominent plant species in GenBank Release 202

PLN non WGS (3.6 million)	EST (25 million total) ^a	TSA (11 million total) ^a
<i>Arabidopsis thaliana</i> 183,213	<i>Zea mays</i> 2,019,524	<i>Triticum aestivum</i> 834,560
<i>Oryza sativa</i> 144,420	<i>Arabidopsis thaliana</i> 1,529,700	<i>Panicum trichoides</i> 477,845
<i>Zea mays</i> 130,002	<i>Glycine max</i> 1,461,723	<i>Pseudotsuga menziesii</i> 375,963
<i>Pinus taeda</i> 115,514	<i>Triticum aestivum</i> 1,286,216	<i>Cuscuta pentagona</i> 275,368
<i>Vitis vinifera</i> 115,294	<i>Oryza sativa</i> 1,253,683	<i>Artemisia annua</i> 265,440
<i>Oryza sativa</i> ^b 90,687	<i>Oryza sativa</i> ^b 987,327	<i>Amaranthus tricolor</i> 237,342
<i>Glycine max</i> 90,019	<i>Hordeum vulgare</i> 828,547	<i>Camelina sativa</i> 210,967
<i>Malus domestica</i> 68,603	<i>Panicum virgatum</i> 720,590	<i>Allium cepa</i> 209,510
<i>Medicago truncatula</i> 68,185	<i>Hordeum vulgare</i> ^c 718,147	<i>Medicago sativa</i> 207,522
<i>Gossypium hirsutum</i> 55,726	<i>Brassica napus</i> 643,944	<i>Elodea nuttallii</i> 206,926

^aThese columns indicate the number of WGS or TSA “master” records, each of which contains hundreds to many thousands of individual contig sequences

^bJaponica cultivar-group

^c*Hordeum vulgare* subsp. *vulgare*

These data include both the WGS contigs as well as genomic scaffolds assembled from the WGS contigs, and such scaffolds are available in the **CON** division of GenBank. Another 17 % of the data are either expressed sequence tags (**EST** division), genome survey sequences (**GSS** division), other high-throughput sequences (**HTG** division) or sequences from transcriptome shotgun assembly projects (**TSA** division). A listing of prominent plant species in the **PLN**, **EST**, and **TSA** divisions is provided in Table 2.

Another increasingly important source of sequence data for plants is next-generation sequencing projects that deposit data into the NCBI Sequence Read Archive (SRA). While SRA [6] is not formally part of GenBank, the sequence reads it contains may be assembled into larger sequences or alignments that can be deposited into GenBank (for example, into the **TSA** division).

In addition to the high-throughput sequences mentioned above, NCBI encourages the submission of sequencing data ranging in complexity from a transcript sequence annotated with a single coding region, to sets of aligned sequences supporting population or phylogenetic studies, or large scale genomic assemblies with detailed annotations.

2.1.4 Submitting Sequence Records to GenBank

Virtually all records enter GenBank as direct electronic submissions, with the majority of authors using the BankIt or Sequin programs described on the GenBank submission Web page (www.ncbi.nlm.nih.gov/genbank).

ncbi.nlm.nih.gov/genbank/submit/). Most journals require authors with sequence data to submit the data to a public database as a condition of publication. GenBank staff can usually assign an accession number (*see Note 3*) to a sequence submission within two working days of receipt. The accession number serves as confirmation that the sequence has been submitted and can be used to retrieve the data when it appears in the database. Direct submissions receive a quality assurance review that includes checks for vector contamination, proper translation of coding regions, correct taxonomy and correct bibliographic citations. A draft of the GenBank record is passed back to the author for review before it enters the database, and authors may ask that their sequences be kept confidential until the time of publication. Since GenBank policy requires that deposited sequence data be made public when the sequence or accession number is published, authors are instructed to inform GenBank staff of the publication date of the article in which the sequence is cited in order to ensure a timely release of the data. Although only the submitting scientist is permitted to modify sequence data or annotations, all users are encouraged to report lags in releasing data or possible errors or omissions to GenBank by writing to update@ncbi.nlm.nih.gov.

About a third of author submissions are received through BankIt (www.ncbi.nlm.nih.gov/WebSub/?tool=genbank), a Web-based data submission tool. Using BankIt, authors enter sequence information and biological annotations, such as coding regions or mRNA features, directly into a series of tabbed forms that allow the submitter to describe the sequence further without having to learn formatting rules or controlled vocabularies. Additionally, BankIt allows submitters to upload source and annotation data using tab-delimited tables. Before creating a draft record in the GenBank flat file format for the submitter to review, BankIt validates the submissions by flagging many common errors and checking for vector contamination using a variant of BLAST called Vecscreen. Help using BankIt, as well as example submission scenarios, is available in the GenBank Submissions Handbook (<http://www.ncbi.nlm.nih.gov/books/NBK51157/>).

NCBI also offers a standalone submission program called Sequin (www.ncbi.nlm.nih.gov/Sequin/index.html) that can be used interactively with other NCBI sequence retrieval and analysis tools. Sequin handles simple sequences such as a cDNA, as well as segmented entries, phylogenetic studies, population studies, mutation studies, environmental samples and alignments. Sequin offers complex annotation capabilities and contains a number of built-in validation functions for quality assurance. In addition, Sequin is able to accommodate large chromosome-scale sequences and read in a full complement of annotations from simple tables. Once a submission is completed, submitters can e-mail the Sequin file to gb-sub@ncbi.nlm.nih.gov. Versions of Sequin for common

computer platforms are available via anonymous FTP (<ftp.ncbi.nlm.nih.gov/sequin>).

NCBI works closely with sequencing centers to ensure timely incorporation of bulk data into GenBank for public release. Submitters of large, heavily annotated genomes may find it convenient to use ‘tbl2asn’ (www.ncbi.nlm.nih.gov/Sequin/table.html), to convert a table of annotations generated from an annotation pipeline into an ASN.1 record suitable for submission to GenBank. Special procedures for the batch submission of EST and GSS sequences are described on the GenBank submission page (www.ncbi.nlm.nih.gov/Genbank/submit.htm). WGS and TSA projects can be uploaded to GenBank using the NCBI submission portal (submit.ncbi.nlm.nih.gov). The tbl2asn output file can be uploaded directly through the portal along with appropriate project and sample metadata. In addition, FASTA and AGP files can be submitted directly for WGS.

2.1.5 Annotations Found in GenBank Records

Each GenBank entry includes a concise description of the sequence, the scientific name and taxonomy of the source organism, bibliographic references, and a table of biological features (*see Note 4*). Annotation is best for GenBank records in the **PLN** division, while records in divisions such as **EST**, **GSS** and **TSA** contain either minimal annotation or no annotation at all.

2.1.6 Integration of GenBank Data with Other Resources

Given the increasing amount of data arising from high-throughput sequencing methods, NCBI has developed a suite of four related resources—BioProject, Genome, Assembly and BioSample—that aggregate these data around four central concepts. The BioProject database [7] contains records that represent funding initiatives for a wide variety of genomic projects. Each record contains information about the project itself and provides links to any data that the project has submitted to NCBI. For genome sequencing projects focused on a single species, the Genome database [4] collects all data, ranging from short reads to fully assembled chromosomes, produced by such projects for that species. The Genome record for a species also will contain sequence data for any sub-species or strains, along with organelle genome sequences for the species. Currently there are over 600 Genome records for plants, almost 50 of which represent complete genomes. Many modern genomic data sets, particularly from higher eukaryotes, represent the genome as a collection of individual chromosome sequences called an assembly. These assemblies are often updated over time, with each update labeled as a unique version. The NCBI Assembly database collects the sequences that comprise an individual version of a genome assembly, along with associated metadata for that assembly. There are currently over 170 assemblies for more than 100 plant species. Finally, the BioSample database [7] collects information about the specimen used as the source of data submitted to

NCBI. Each of these four databases links directly to relevant GenBank data and, as discussed below, offers a unique path to search and retrieve these data dependent on the user's goals.

For plant species with genome mapping data or genome annotations, NCBI may provide graphical views of the genomic maps in the NCBI Map Viewer or records in the Gene database. Currently over 120 plant species have graphical maps available and more than 450 have Gene records. When available, GenBank records are shown aligned to the genomic maps, and will be linked as supporting data to Gene records. Sequence variations may also be displayed in the Map Viewer, and over 25 million SNPs have been mapped to plant genomes, the vast majority to genomes of *Glycine max*, *Oryza sativa*, *Sorghum bicolor* and *Arabidopsis thaliana*. In addition, the more than 50 plant species that have more than 70,000 EST sequences in GenBank have been incorporated into the UniGene database [4], where these ESTs are combined with other transcript sequences in GenBank and partitioned into over 1.3 million gene-oriented clusters. Links from UniGene are also available to Gene, HomoloGene and Protein where possible. As a consequence, Entrez (Subheading 2.2.1) can be used to match a GenBank EST accession number (see Note 3) to a gene location, a protein sequence, and homologous genes in many organisms.

2.2 Accessing GenBank

GenBank data can be accessed in several ways. The Entrez system on the NCBI web site allows users to search, view and download any arbitrary subset of GenBank, while the entire database can be downloaded from the NCBI FTP site (<ftp.ncbi.nlm.nih.gov>). In addition, programmers can access GenBank data using the Entrez Programming Utilities (E-Utilities), the public API to the Entrez system (<eutils.ncbi.nlm.nih.gov>).

2.2.1 Interactive Access with Entrez

The sequence records in GenBank are accessible using Entrez [4], a robust and flexible database retrieval system that covers over 40 biological databases containing almost a billion individual records ranging from DNA and protein sequences to genome maps, literature abstracts in PubMed, full text articles in PMC, gene expression data in GEO [8], variations in SNP and dbVar [9, 10], the full NCBI taxonomy [5], protein domains and 3D structures [11, 12], chemicals in PubChem [13, 14] and many other data types. GenBank data are found in three Entrez databases: the EST and GSS databases contain all sequences in the EST and GSS GenBank divisions, respectively, while the Nucleotide database contains sequences from all other GenBank divisions (as well as sequences from databases other than GenBank). The GenBank data may be selectively accessed within Entrez using query limitations (see Note 5). Conceptual translations of coding regions annotated on GenBank sequences are available in the Protein database.

Records within the Entrez system are linked to other records both within and across databases. An example of a simple linkage is that between a GenBank sequence record and the PubMed abstract for the paper listed in the ‘Journal’ section (*see Note 2*) of the GenBank record. Computational linkages are also made between nucleotide and protein sequences, such as those based on sequence similarities discovered using BLAST [15, 16]. In addition, records available in Entrez may offer LinkOuts (www.ncbi.nlm.nih.gov/projects/linkout/) that lead to a variety of external databases. Queries on the Entrez databases are made with simple text words combined using boolean logic and either limited to a particular record field (*see Note 5*) or applied to all fields. A web-based service called Batch Entrez allows bulk sequence downloads specified by an arbitrary set of GenBank identifiers supplied from a local file.

2.2.2 Scripted Access Through Entrez with the Entrez Programming Utilities

Entrez queries of GenBank and downloads of individual records or sets of records may be made through the Entrez system from scripts using a set of server-side utilities called the Entrez Programming Utilities (*see Note 6*). Full documentation for these utilities is available at eutils.ncbi.nlm.nih.gov.

2.2.3 Bulk Downloads of GenBank via FTP

The full bimonthly GenBank release and the daily updates, which also incorporate sequence data from ENA and DDBJ, are available by anonymous FTP from NCBI in both flatfile and ASN.1 formats (<ftp://ftp.ncbi.nlm.nih.gov/genbank/>, <ftp://ftp.ncbi.nlm.nih.gov/ncbi-asn1/>). The GenBank data are also available at a mirror site at Indiana University (<ftp://bio-mirror.net/biomirror/genbank/>). As described in Subheading 2.1.1, the full release in flatfile format is distributed as a set of compressed files. To download GenBank files using a command line FTP client, connect to <ftp.ncbi.nlm.nih.gov>, log in as ‘anonymous’ and give your email address as password.

For many purposes a download of the entire GenBank databases is not required since standard sequence-similarity searches, such as BLAST, may be performed remotely on the GenBank data at NCBI. However, if a local copy of GenBank is required, one major consideration is local storage space.

As of GenBank release 202, the uncompressed GenBank flatfiles require 642 gigabytes of disk space. An alternative option is to download the ASN.1 format data, which requires 538 gigabytes once uncompressed. Once a full release of GenBank had been saved locally, it can be kept current using incremental updates. For this purpose, NCBI provides a noncumulative set of updates at <ftp.ncbi.nlm.nih.gov/genbank/daily-nc>. A Perl script at <ftp.ncbi.nlm.nih.gov/genbank/tools/> converts a set of daily updates into a cumulative update (<ftp.ncbi.nlm.nih.gov/genbank/tools/>).

3 Methods

Four general methods that are central to making use of GenBank include downloads of all or a subset of the database to support local analysis, the construction and use of local GenBank-derived databases for sequence similarity searches, and the execution of remote searches of the database using Web or command line clients. These methods are discussed in the sections that follow.

3.1 Download All GenBank PLN Division Sequences

3.1.1 Strategy

The GenBank **PLN** division is a source of well-annotated sequences that can serve as a compact, information-rich local database. For such simple, division-oriented bulk downloads, FTP transfers are the most convenient given that the division name forms part of the file names on the GenBank FTP site. Downloading these files is simply a matter of connecting to the NCBI FTP site and specifying files of the name '**gbpln*.seq.gz**' (where the asterisk is a wild card that matches any set of characters). The download will result in a local set of compressed files for sequence records in the GenBank flatfile format. To view and use the records, they must be uncompressed.

3.1.2 Execution

To begin, perform an anonymous FTP login to the NCBI FTP server using either a command line FTP client or a web browser. If using a command line FTP client, connect to <ftp.ncbi.nlm.nih.gov> and then issue the following commands:

1. **cd genbank**
2. **get gbpln*.seq.gz**
3. Following the completion of the transfers, type '**quit**' at the FTP prompt.

Although less convenient, these files can be downloaded using a web browser as follows:

1. Navigate to <ftp://ftp.ncbi.nlm.nih.gov/genbank>
2. Click on each **gbpln*.seq.gz** file in turn to download it.

As described in Subheading 2.1.1, the GenBank data are also available in the compact and versatile ASN.1 format. Besides their significantly smaller size, the ASN.1 format files offer other advantages over the compressed flatfiles. Using a suite of command line tools available from NCBI (*see Note 7*), ASN.1 files can be used to generate records in a variety of other formats. These formats include the GenBank flatfile (*see Note 2*), FASTA, 5-column Feature Table (*see Note 8*) and INSDC XML. In addition, both the nucleotide sequences, and the protein sequences derived from their coding sequence (CDS) annotations are readily accessible. ASN.1 formatted GenBank records can also be used to generate databases, both nucleotide and protein, for local analysis using BLAST (Subheading 3.4).

3.2 Download a Set of GenBank Sequences for a Single Plant Species

3.2.1 Strategy

A method to download a complete set of sequences for a single plant species, such as *Vitis vinifera*, must retrieve all *Vitis vinifera* sequences regardless of GenBank division. In addition to the **PLN** division, divisions such as **EST**, **STS**, and **GSS** also contain plant sequences. Rather than downloading each division in its entirety merely to get the *Vitis vinifera* data, it is more practical to use the flexibility of the Entrez search and retrieval system. Using Entrez, one can specify the subset of GenBank to download, choose a download format, and then download the sequence records as a single batch.

3.2.2 Execution

As described in Subheading 2.2.1, GenBank sequences are contained in three Entrez databases: Nucleotide, EST and GSS. To retrieve all *Vitis vinifera* sequences in GenBank, one therefore needs to retrieve data from each of these three databases. To begin, first retrieve all sequences for *Vitis vinifera* using the following query in the Nucleotide database:

vitis vinifera[orgn]

As shown in Fig. 1, links to the results from the EST and GSS databases are provided above the search results. Download these data by following those two links and using the ‘Send to’ menu. To download the data in the Nucleotide database, click on the INSDC limit in the upper right to restrict the set to GenBank data, and then use the ‘Send to’ menu as with EST and GSS. The following Entrez query accomplishes the same GenBank restriction:

vitis vinifera[orgn] AND srcdb ddbj/embl/genbank[prop]

As of May 2014, these queries retrieved 247,000 *Vitis vinifera* sequences from Nucleotide, 447,000 from EST and 229,000 from GSS for a total of 923,000 sequences. While it is possible to

Fig. 1 Portion of the search results page in Entrez Nucleotide with the query ‘vitis vinifera[orgn]’. Links above the list of results allow access to the records retrieved in the EST and GSS databases, while filters in the *upper right* (such as INSDC) allow the results to be further restricted

download such a set using a web browser, an alternative approach is to use the Entrez Programming Utilities to accomplish the download (*see* **Note 6**).

3.3 Download the Complete Genome of a Plant Species

3.3.1 Strategy

In recent years the number of plant species with sequenced genomes has continued to increase. As of this writing, about 60 plant species have a genome assembly that contains assembled chromosomes (with or without gaps), while another 80 have at least scaffold assemblies. As these data continue to mature, the NCBI interfaces will consequently continue to adapt to accommodate these changes; however, the Genome database should remain a central place for users to access genomic datasets for a given species. Another, and often more direct, approach to accessing these datasets is to download them from the NCBI FTP site, and this approach will be discussed here.

3.3.2 Execution

There are two primary areas of the NCBI FTP site in which eukaryotic genome datasets may be found: the GenBank genomes site (<ftp.ncbi.nlm.nih.gov/genbank/genomes/>) and the RefSeq genomes site (<ftp.ncbi.nlm.nih.gov/genomes/>). Each of these sites contains a list of subdirectories with names corresponding to the scientific names of each species. Once within the directory for the desired species, 'readme' files will describe the contents, which may vary somewhat from species to species depending on the nature of that species' genome and the methods used in the sequencing project. In general, separate sequence files will be available for each chromosome, and therefore downloading the genome is simply a matter of downloading the individual chromosome data files.

3.4 Establish and Perform Sequence Similarity Searches on a Local Database of PLN Division Sequences

3.4.1 Strategy

The Basic Local Alignment Search Tool, or BLAST, is the most widely used program for sequence similarity searches. In addition to comparing nucleotide sequences, BLAST can also translate nucleotide sequences into all six reading frames at run time in order to compare protein coding regions. There are three basic approaches to performing BLAST searches against NCBI sequence databases: (1) using the NCBI BLAST web interface, (2) using a local installation of BLAST but using the databases on NCBI servers; (3) using a local installation of BLAST with local databases. BLAST binaries for standard computing platforms are available for download on the NCBI FTP site (<ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/>). While a complete discussion of the BLAST algorithm and the interpretation of results is beyond the scope of this chapter, details can be found elsewhere [15–19]. Comprehensive documentation on the BLAST executables is also available on the NCBI Bookshelf (<http://www.ncbi.nlm.nih.gov/books/NBK1763/>).

3.4.2 Execution

To search PLN division sequences on the NCBI web interface using BLAST (blast.ncbi.nlm.nih.gov), it is simply a matter of loading the nucleotide blast page and restricting the default database (nr/nt) to the desired organism (or organisms) using the Organism selection box, and then adding the Entrez query ‘srcdb ddbj/embl/genbank[prop]’ (*see* Subheading 3.2.2). If many such BLAST searches need to be performed, it may be advantageous to download local BLAST binaries so that the searches can be automated. The BLAST binaries allow searches to access the databases on NCBI servers, obviating the need for a local copy and ongoing updating of these (often large) files. The following command will run a standard nucleotide BLAST search limited to green plant sequences in the INSDC databases:

```
blastn -db nt -query myfile_nuc -out myoutfile -entrez_query  
"viridiplantae[orgn]ANDsrcdbddbj/embl/genbank[prop]"  
-task blastn -remote
```

In the above command, the parameter ‘-db’ specifies by the name of the database, and ‘-query’ is followed by the name of a file containing one or more FASTA formatted sequences to be used as queries. The ‘-out’ parameter is given the name of the desired output file, and ‘-entrez_query’ the Entrez query used to restrict the given database. The **task** option specifies the algorithm used; in this case **blastn** invokes standard nucleotide BLAST. The default value is ‘megablast’, a faster but less sensitive version of nucleotide BLAST that is useful for finding matches within the same or closely related species [20]. Finally, the ‘-remote’ flag causes **blastn** to access databases on the NCBI servers rather than local databases.

To run BLAST searches independently of NCBI servers, both the BLAST executable and the desired sequences (such as the GenBank sequences downloaded using Subheadings 3.1 and 3.2 above) need to be downloaded to a local machine. Once this is done, the first task in executing a BLAST search is to convert the sequence data into a local BLAST database. This is accomplished by the **makeblastdb** program, which can create a local BLAST database from a file of concatenated FASTA format sequences, or from an ASN.1 format GenBank file. The following command line is used to create a local nucleotide database from a file of concatenated FASTA format sequences contained within the file ‘myfasta’:

```
makeblastdb -in myfasta -input_type fasta -dbtype nucl  
-parse_seqids
```

The **parse_seqids** flag causes **makeblastdb** to create indices that allow individual sequence records to be retrieved by another program in the BLAST package, **blastdbcmd** (*see* Note 9), on the basis of sequence identifiers found in the definition lines of the records.

To create a nucleotide sequence database with a set of binary ASN.1 format GenBank sequence files, use the following:

```
makeblastdb -in gbpln1.aso -input_type asn1_bin -dbtype  
nucl -parse_seqids
```

To create a protein sequence database from the corresponding translations of annotated CDS features on the nucleotide sequences contained in **gbpln1.aso**, use the following:

```
makeblastdb -in gbpln1.aso -input_type asn1_bin -dbtype  
prot -parse_seqids
```

By default **makeblastdb** will produce several files whose names consist of the input file name (-in) followed by one of various extensions. To search this database, simply provide the name of the input file provided to **makeblastdb**. For example, to search the nucleotide database formatted above using the '**blastn**' program with a nucleotide query sequence in FASTA format within a file named '**myfile_nuc**', use the following:

```
blastn -query myfile_nuc -db gbpln1.aso -task blastn -out  
myoutfile
```

To search the protein translations arising from the CDS features on the records in **gbpln1.aso**, assuming that the protein sequence version of **gbpln1.aso** has been created using **makeblastdb** as described above, use:

```
blastp -query myfile_prot -db gbpln1.aso -out myoutfile -  
evalue 1e-6
```

A nucleotide query can also be used to query the protein translations using the **blastx** algorithm, which will translate the query into all six reading frames:

```
blastx -query myfile_nuc -db gbpln1.aso -out myoutfile -  
evalue 1e-6
```

The '**-db**' parameter in these commands is followed by the name of the database formatted using **makeblastdb**. The quality of the alignments returned by BLAST can be controlled using the '**-evalue**' parameter to set an 'expect value' limit. In this case, an expect value of 1e-6 (0.000001) has been specified, which should exclude those alignments expected to occur by chance more than 0.000001 times in a database of the size of **gbpln1.aso**.

A large number of parameters may be specified on the various BLAST executables, and these parameters determine the number, format, and quality of the alignments returned. To see them all, type '**-help**' after any of the executable names. For detailed documentation on the parameters, see the online documentation at www.ncbi.nlm.nih.gov/books/NBK1763/.

3.5 Identify Potential Coding Regions in TSA Datasets for Green Plants

3.5.1 Strategy

In the past few years the TSA division of GenBank has been one of the most rapidly expanding divisions. TSA data are a more valuable version of raw next-generation sequencing data, such as SRA, because they have been partially assembled and therefore are more likely to represent a full transcript. For plant species with little assembled genomic data, TSA data can therefore be a rich collection for identifying putative protein coding regions (CDS) in species without genome assemblies. By using the translated BLAST algorithm *tblastn*, one can search a protein query against a TSA database translated into all six reading frames, easily revealing these putative CDS sequences.

3.5.2 Execution

To search the TSA data for CDS regions corresponding to a given protein (or proteins), first assemble the query protein sequences either as accession numbers or FASTA files. As with Subheading 3.4 above, BLAST searches against TSA data can be performed on the NCBI web interface or using local BLAST installations, with or without local copies of the TSA database. On the NCBI web interface, the TSA database is one of the options on the *tblastn* search page, which also allows the database to be limited by organism and/or Entrez query (*see* Subheading 3.4.2). Figure 2 shows the

Sequences producing significant alignments:

Select: All None Selected:0

Alignments Download GenBank Graphics

Description	Max score	Total score	Query cover	E value	Ident	Accession
<input type="checkbox"/> Cajanus cajan cultivar Asha contig02520 Asha mRNA sequence	917	917	100%	0.0	90%	E2650384.1
<input type="checkbox"/> Cajanus cajan cultivar UPAS 120 contig22788 UPAS mRNA sequence	891	891	96%	0.0	90%	EZ840505.1
<input type="checkbox"/> Lotus corniculatus CL2124 Contig2 mRNA sequence	874	874	99%	0.0	84%	GACB01018339.1
<input type="checkbox"/> Pisum sativum peapol_c951 transcribed RNA sequence	872	872	99%	0.0	84%	GAMJD1000949.1
<input type="checkbox"/> Medicago sativa comp57474_c0_seq1 mRNA sequence	866	866	99%	0.0	83%	GAFF01041300.1
<input type="checkbox"/> Arachis duranensis DurSNP_c3300 Ardu mRNA sequence	801	801	94%	0.0	80%	HP003292.1
<input type="checkbox"/> Medicago sativa comp57474_c0_seq2 mRNA sequence	794	794	90%	0.0	85%	GAFF01041301.1
<input type="checkbox"/> Ficus carica mRNA contig: FICAF00359 mRNA sequence	772	772	97%	0.0	75%	EX377333.1
<input type="checkbox"/> Betula platyphylla Unigene41002_vs_RNA mRNA sequence	755	755	99%	0.0	72%	jP708789.1
<input type="checkbox"/> Corvus avellana CAV_005301.ca mRNA sequence	755	755	99%	0.0	71%	KA388644.1
<input type="checkbox"/> Vitis vinifera Locus_8299_Transcript_3/3_Confidence_0.500 Vvvi mRNA sequence	749	749	99%	0.0	71%	KA156834.1
<input type="checkbox"/> Vitis vinifera Grain_1_3_1713_Transcript_4_4_Confidence_0.500_Length_2174 transcribed RNA sequence	748	748	99%	0.0	71%	GAKHD1027889.1
<input type="checkbox"/> Humulus lupulus comp74973_c0_seq1 mRNA sequence	739	739	98%	0.0	73%	GAAW01063355.1
<input type="checkbox"/> Momordica charantia Locus_9511_Transcript_2/2_Confidence_0.829_Length_2104 transcribed RNA sequence	741	741	97%	0.0	71%	GANFD1018973.1
<input type="checkbox"/> Momordica charantia Locus_13804_Transcript_2/2_Confidence_0.727_Length_2105 transcribed RNA sequence	739	739	97%	0.0	71%	GANG01047922.1
<input type="checkbox"/> Mangifera indica com8165_c0_seq1 transcribed RNA sequence	731	731	97%	0.0	71%	GBCV01007506.1
<input type="checkbox"/> Betula platyphylla Unigene42177_All Beptwood mRNA sequence	755	1510	99%	0.0	72%	KA201189.1
<input type="checkbox"/> Paeonia suffruticosa CL7248.Contig1 mRNA sequence	733	733	99%	0.0	70%	GANK01000150.1
<input type="checkbox"/> Camptotheca acuminata caa10501_iso2 mRNA sequence	735	735	98%	0.0	71%	GACFD1047004.1
<input type="checkbox"/> Arachis hypogaea Unigene29063 mRNA sequence	724	724	84%	0.0	82%	GAER01020945.1
<input type="checkbox"/> Gossypium hirsutum Tm1_Mira454_Contig_3328 transcribed RNA sequence	721	721	97%	0.0	70%	GALV01025840.1
<input type="checkbox"/> Vitis vinifera Skin_2_3_6082_Transcript_1_4_Confidence_0.778_Length_2294 transcribed RNA sequence	728	728	99%	0.0	64%	GAKHD1096838.1

Fig. 2 Portion of the results page of a *tblastn* search against the TSA database restricted to green plants (*viridiplantae[orgn]*). The query used was NP_001236858, the sequence for ACC synthase from soybean. The results contain hits from a variety of plant species, and as shown the hits have sequence identities with the query from 64 to 90 %, and all but two cover at least 97 % of the query sequence

best hits resulting from such a search using a soybean protein (NP_001236858) as query. To run `tblastn` locally but against the databases on NCBI servers, use the following command:

```
tblastn -db tsa_nt -query myfile_prot -out myoutfile -entrez_query "viridiplantae[orgn]" -remote
```

As mentioned above, this approach has distinct advantages: (1) there is no need to download and update a local copy of a large database (as of this writing the TSA collection requires 17 GB of disk space); and (2) organism and Entrez limitations to the database can be applied at runtime. On the other hand, the method will be limited by network and NCBI traffic, and so it may be desirable to download the preformatted TSA database to local machines. Because of its size, the TSA database on the NCBI ftp site is split into several volumes, each approximately 1 GB of compressed data. To facilitate downloading such sets of files, NCBI provides a utility script named `update_blastdb.pl` as part of the BLAST software package. This script will download all of the component files of the TSA database (or other preformatted databases) to local disk, where they can then be uncompressed and extracted. Because these files are preformatted, running `makeblastdb` is unnecessary, and the files can be immediately used in a search:

```
tblastn -query myfile_prot -db tsa_nt -out myoutfile -eval 1e-6
```

As with all local BLAST executables, it is possible to restrict the TSA database to only those sequences matching a GI list in a local file (with one GI per line in the file). For example, running the following query in the Nucleotide database and then downloading the data as a GI list will create such a file that can accomplish an organism restriction to green plants:

```
tsa[keyword] AND viridiplantae[orgn]
```

If this file is named `greenplantTSA.gi`, then the following search will be restricted to green plants:

```
tblastn -query myfile_prot -db tsa_nt -out myoutfile -eval 1e-6 -gilist greenplantTSA.gi
```

While the above approach for restricting the search is valid at the time of this writing, these functions will be maturing over time, and users are encouraged to visit the BLAST web site regularly for updates and improvements to these methods.

4 Notes

1. The files in the GenBank releases are partitioned into 19 'divisions' that correspond roughly to taxonomic groups such as bacteria (**BCT**), viruses (**VRL**), primates (**PRI**), and rodents

(**ROD**). Additional divisions have been added over time to support specific sequencing strategies. These include divisions for expressed sequence tag (**EST**), genome survey (**GSS**), high throughput genomic (**HTG**), high throughput cDNA (**HTC**), environmental sample (**ENV**) and transcript shotgun assembly (**TSA**) sequences. To facilitate downloads, most divisions are partitioned into multiple files for the bimonthly GenBank releases on NCBI's FTP site. In addition, three special classes of records exist that do not appear within the usual 19 divisions of GenBank: WGS, TPA and TSA records. Over 600 billion bases of WGS sequence appear in GenBank as sets of WGS contigs, many of them bearing annotations, originating from a single sequencing project. Third Party Annotation (TPA) (www.ncbi.nih.gov/Genbank/TPA.html) records support the reporting of published, experimentally confirmed sequence annotation by a scientist other than the original submitter of the primary sequence. TSA records are assembled from short reads (from SRA or the Trace Archive) or from ESTs. The content of the GenBank divisions is summarized in Table 3.

2. The GenBank flatfile is the standard data format used for GenBank records and is the format of the data in the GenBank FTP files. Each record begins with a LOCUS line followed by a header containing the database identifiers, the title of the record, references, and submitter information. The header is followed by the feature table (*see Note 4*) and the sequence itself on the line following the 'origin' field. The '/' symbol in the FTP files marks the boundary between successive records. The GenBank flatfile is described in detail in the GenBank release notes (<ftp.ncbi.nlm.nih.gov/genbank/gbrel.txt>). In the Entrez system, the GenBank format is the default display for records in the traditional divisions. An interactive sample record is linked from the GenBank home page (www.ncbi.nlm.nih.gov/genbank/).
3. Each GenBank record, consisting of both a sequence and its annotations, is assigned a unique identifier called an 'accession number' that is shared across the three INSDC members (GenBank, DDBJ, ENA). The accession number appears on the ACCESSION line of a GenBank record and remains constant over the lifetime of the record, even when there is a change to the sequence or annotation. Changes to the sequence data itself are tracked by an integer extension of the accession number, and this Accession.version identifier appears on the VERSION line of the GenBank flat file. An entry appearing in the database for the first time has an 'Accession.version' identifier equivalent to the ACCESSION number of the GenBank record followed by '.1'. In addition, each version of the sequence is assigned a unique integer identifier called a 'GI

Table 3
Division codes and content of the 19 GenBank divisions

Code	Description
Traditional GenBank Divisions	
BCT	Bacterial sequences
PRI	Primate sequences
MAM	Other mammalian sequences
VRT	Other vertebrate sequences
INV	Invertebrate sequences
PAT	Patent sequences
PLN	Plant, fungal, and algal sequences
VRL	Viral sequences
PHG	Bacteriophage sequences
SYN	Synthetic and chimeric sequences
UNA	Unannotated sequences, including some WGS sequences obtained via environmental sampling methods
Nontraditional GenBank Divisions	
EST	EST division sequences, or expressed sequence tags, are short single pass reads of transcribed sequence. Over 25 million ESTs are derived from almost 1000 plant species.
STS	STS division sequences include anonymous STSs based on genomic sequence as well as gene-based STSs derived from the 3' ends of genes and ESTs. STS records usually include primer sequences, annotations, and PCR reaction conditions. About 160,000 of the STS sequences in GenBank are of plant origin.
GSS	GSS records are predominantly single reads from Bacterial Artificial Chromosomes ('BAC-ends') used in a variety of genome sequencing projects. GSS records for plant species number over 16 million.
ENV	The ENV division of GenBank, for non WGS sequences obtained via environmental sampling methods in which the source organism is unknown, debuted with release 147 in April 2005.
HTG	The HTG division of GenBank contains unfinished large-scale genomic records that are in transition to a finished state. These records are designated as Phase 0-3 depending on the quality of the data. Upon reaching Phase 3, the finished state, HTG records are moved into the appropriate taxonomic division of GenBank.
HTC	The HTC division of GenBank accommodates high-throughput cDNA sequences. HTCs are of draft quality but may contain 5' UTRs and 3' UTRs, partial coding regions, and introns.
CON	Large records that are assembled from smaller records, such as eukaryotic chromosomal sequences or WGS scaffolds, are represented in the GenBank 'CON' division. CON records contain sets of assembly instructions to allow the transparent display and download of the full record using tools such as NCBI's Entrez.
TSA	Transcriptome shotgun data are assembled from sequences deposited in the NCBI Trace Archive, the Sequence Read Archive (SRA) and the EST division of GenBank.

number' that also appears on the VERSION line of GenBank flatfile records:

ACCESSION AF000001

VERSION AF000001.1 GI: 987654321

When a change is made to a sequence in a GenBank record, a new GI number is issued to the updated sequence and the version extension of the Accession.version identifier is incremented. The accession number for the record remains unchanged, and will always retrieve the most recent version of the record; the older versions remain available under the old Accession.version identifiers and their original GI numbers. The Revision History report, available from the 'Display Settings' menu on the sequence record view, summarizes the various updates for that GenBank record.

A similar system tracks changes in the corresponding protein translations. These identifiers appear as qualifiers for CDS features in the FEATURES portion of a GenBank entry, e.g., /**protein_id='AAA00001.1'** Protein sequence translations also receive their own unique gi number, which appears as a second qualifier on the CDS feature:

/db_xref='GI:1233445'

4. The feature table is the portion of the GenBank record that provides information about the biological features annotated on the nucleotide sequence. These features include coding regions and their protein translations, noncoding regions, genes, variations, sequence tagged sites, transcription units, repeat regions, and sites of mutations or modifications. The International Sequence Database Collaboration (www.insdc.org) produces a document describing and identifying the features allowed on GenBank, DDBJ and ENA records (<http://www.insdc.org/documents/feature-table>).
5. The GenBank database and protein sequences arising from coding sequence annotations on GenBank records can be searched at NCBI using BLAST using either a web interface or a command-line client. In either case, subsets of the data may be selected for searching using Entrez (Subheading 2.2.1) queries. A query used to limit a search to sequences from a particular organism has the form 'organism[orgn]' where 'organism' is an organism name and the search is limited to terms indexed within the Entrez 'organism' field by specifying 'orgn' within square brackets. For example, to specify only sequences from *Arabidopsis thaliana*, use '**Arabidopsis thaliana[orgn]**'. Some Entrez queries involving terms indexed in the Entrez '**properties**' field are listed in Table 4. Entrez queries can be combined using boolean operators such as

Table 4
Entrez queries that are useful in limiting BLAST searches

Terms in the Entrez 'properties' field	Effect	Example of use
'gbdiv X' where X is one of pri, rod, mam, vrt, inv, pln, bct, vrl, phg, syn, una, est, pat, sts, gss, htg, htc, env	Limit to sequences within a GenBank division	gbdiv pln[prop]
'biomol' X where X is one of crna, genomic, genomic mrna, other, pre mrna, rna,rrna, scrna, snorna, snrna, transcribed rna, trna	Limit to a molecule type	biomol mrna[prop]
'srcdb X' where X is one of ddbj, ddbj/embl/genbank, embl, genbank, pdb, tpa ddbj, tpa ddbj/embl/genbank, tpa embl, tpa genbank, refseq ^a	Limit to source database. To limit to GenBank, use 'ddbj/embl/genbank' ^b	srcdb ddbj/embl/genbank[prop]

^aAdditional 'refseq' terms are available but are not shown

^bThe term '**srcdb genbank[prop]**' limits to records within GenBank that entered by submission to GenBank at NCBI. Because the data in GenBank also includes data submitted to the DDBJ and EMBL partners of the INSDC, one must specify '**srcdb ddbj/embl/genbank[prop]**' to include all the records in GenBank

'AND', 'OR', and 'NOT' (Subheading 3.2.2). On the BLAST web pages, Entrez queries are typed into the box labeled "Entrez query".

- The Entrez Programming Utilities (E-utilities) are the public API for the Entrez system and consist of a set of nine server-side programs that allow automated access to the Entrez search and retrieval functions. The E-utilities (eutils.ncbi.nlm.nih.gov) accept a set of parameters that may be URL-encoded or transferred via the SOAP protocol. Searches of Entrez are performed using '**esearch**'; short record summaries are retrieved using '**esummary**'; full records may be downloaded using '**efetch**'; and linking between records may be performed using '**elink**'. Additional E-utilities are available for more specialized functions.

The E-utilities may be used from within any programming language that supports the posting of a URL. Results are returned in XML for all E-utilities except '**efetch**', which supports return modes of XML, HTML, text and ASN.1 as well as return formats such as the GenBank flatfile, FASTA, and the INSDC XML format. Several sample E-utility URLs are shown in Table 5. Additional examples, including sample Perl scripts, are provided in Chapter 4 of the online documentation (eutils.ncbi.nlm.nih.gov).

- NCBI offers command line utilities for working with ASN.1 formatted data. These utilities are available for several platforms and may be downloaded from ftp.ncbi.nlm.nih.gov/asn1-converter/.

Table 5
Representative URLs for Entrez Programming Utility calls

1. Retrieve GI numbers for all GenBank plant sequences esearch.fcgi? db=nucleotide&term=srcdb+ddbj/embl/genbank[Properties]+AND+plants[orgn]
2. Retrieve the XML document summary for GI 42494965 esummary.fcgi? db=nucleotide&id=42494965
3. Retrieve the GenBank flat file with annotations for GI 42494965 efetch.fcgi? db=nucleotide&id=42494965&retmode=text&rettype=gbwithparts
4. Retrieve the GI number for the protein product of GI 5881673 elink.fcgi? db=protein&dbfrom=nucleotide&id=5881673

These URLs should be prefixed with the E-utility base URL: <http://eutils.ncbi.nlm.nih.gov/entrez/eutils/>

Table 6
Selected NCBI utility programs for conversion of data from and to the ASN.1 format

ASN1 Converter	Function
asn2all	Converts GenBank release files in ASN.1 format to a variety of other formats
asn2fsa	Converts binary or text ASN.1 sequence files to FASTA format
asn2gb	Converts binary or text ASN.1 sequence files to GenBank or GenPept flatfile formats
asn2idx	Generates accession/file offset indices for Bioseq-set release files
asn2xml	Converts binary or text ASN.1 sequence files to XML format
asnval	Validates ASN.1 release files
tbl2asn	Automates the creation of sequence records for submission to GenBank by reading feature annotations given in the 5-column feature table format and generating an ASN.1 file

To see a complete list of command line parameters for any of the programs, run the program with a trailing dash and no parameter. A list of several of these programs with brief descriptions is given in Table 6. One particularly useful program is **asn2all**, and some examples of using it follow.

The program **asn2all** is primarily intended to generate reports from the binary ASN.1 Bioseq-set GenBank release files (<ftp://ftp.ncbi.nlm.nih.gov/ncbi-asn1/>).

The following command will generate GenBank flatfile records for the nucleotide sequences as well as GenPept flatfile records for protein sequences contained within **gbpln1.aso** (one of the uncompressed ASN.1 files from the PLN division). These two sets will appear in the files “**gbpln1.nuc**” and “**gbpln1.prt**”, respectively.

```
asn2all -i gbpln1.aso -a t -b T -f g -o gbpln1.nuc -v gbpln1.prt
```


Table 7
Output format options for asn2all

Value of the '-f' flag	Resulting format
g	GenBank (nucleotide) or GenPept (protein)
f	FASTA
t	5-column feature table
s	INSD formatted XML
y	TinySet XML (XML version of FASTA)
a	ASN.1 of entire record
x	XML version of entire record (structured as in the ASN.1 format)

Additional formats can be obtained by changing the value of the “-f” parameter (Table 7). The “-a t” parameter value invokes batch processing of a GenBank release file and “-b T” indicates that the input file is binary ASN.1.

A remote fetching option, “-r T”, allows the download of an ASN.1 record from NCBI over a network connection using an accession number or NCBI ‘gi’ identifier (*see Note 3*). For example, to perform a remote fetch of the feature table within the GenBank record for the *Epifagus virginiana* chloroplast genome (accession number **M81884**) use the following:

```
asn2all -r T -A M81884 -f t
```

This produces output in the 5-column Feature Table format described in **Note 8**.

8. When submitting sequences to GenBank that have annotations, submitters have the option to upload these annotations using a file format commonly referred to as the “5-column Feature Table.” This format specifies a simple text file where the annotation data are entered in tab-delimited columns. Details about this format are provided in the GenBank Submission Handbook (www.ncbi.nlm.nih.gov/books/NBK63592/).
9. The program ‘**blastdbcmd**’ is part of the standalone BLAST package and is a tool for interacting with BLAST databases formatted by **makeblastdb**. For example, **blastdbcmd** can provide basic statistics about a BLAST database and download specific records from that database. Full details of how to use **blastdbcmd** are provided in the BLAST+ documentation (www.ncbi.nlm.nih.gov/books/NBK1763/).

Acknowledgements

Funding for this work was provided by the Intramural Research Program of the National Institutes of Health, National Library of Medicine.

References

- Benson DA, Clark K, Karsch-Mizrachi I, Lipman DJ, Ostell J, Sayers EW (2014) GenBank. *Nucleic Acids Res* 42:D32–D37
- Pakseresht N, Alako B, Amid C, Cerdeno-Tarraga A, Cleland I, Gibson R, Goodgame N, Gur T, Jang M, Kay S et al (2014) Assembly information services in the European Nucleotide Archive. *Nucleic Acids Res* 42:D38–D43
- Kosuge T, Mashima J, Kodama Y, Fujisawa T, Kaminuma E, Ogasawara O, Okubo K, Takagi T, Nakamura Y (2014) DDBJ progress report: a new submission system for leading to a correct annotation. *Nucleic Acids Res* 42:D44–D49
- NCBI Resource Coordinators (2014) Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res* 42:D7–D17
- Federhen S (2012) The NCBI Taxonomy database. *Nucleic Acids Res* 40:D136–D143
- Kodama Y, Shumway M, Leinonen R (2012) The Sequence Read Archive: explosive growth of sequencing data. *Nucleic Acids Res* 40:D54–D56
- Barrett T, Clark K, Gevorgyan R, Gorenkov V, Gribov E, Karsch-Mizrachi I, Kimelman M, Pruitt KD, Resenchuk S, Tatusova T et al (2012) BioProject and BioSample databases at NCBI: facilitating capture and organization of metadata. *Nucleic Acids Res* 40:D57–D63
- Barrett T, Wilhite SE, Ledoux P, Evangelista C, Kim IF, Tomashevsky M, Marshall KA, Phillippy KH, Sherman PM, Holko M et al (2013) NCBI GEO: archive for functional genomics data sets--update. *Nucleic Acids Res* 41:D991–D995
- Lappalainen I, Lopez J, Skipper L, Hefferon T, Spalding JD, Garner J, Chen C, Maguire M, Corbett M, Zhou G et al (2013) DbVar and DGVa: public archives for genomic structural variation. *Nucleic Acids Res* 41:D936–D941
- Sherry ST, Ward MH, Kholodov M, Baker J, Phan L, Smigielski EM, Sirotkin K (2001) dbSNP: the NCBI database of genetic variation. *Nucleic Acids Res* 29:308–311
- Madej T, Address KJ, Fong JH, Geer LY, Geer RC, Lanczycki CJ, Liu C, Lu S, Marchler-Bauer A, Panchenko AR et al (2012) MMDB: 3D structures and macromolecular interactions. *Nucleic Acids Res* 40:D461–D464
- Marchler-Bauer A, Lu S, Anderson JB, Chitsaz F, Derbyshire MK, DeWeese-Scott C, Fong JH, Geer LY, Geer RC, Gonzales NR et al (2011) CDD: a Conserved Domain Database for the functional annotation of proteins. *Nucleic Acids Res* 39:D225–D229
- Wang Y, Xiao J, Suzek TO, Zhang J, Wang J, Zhou Z, Han L, Karapetyan K, Dracheva S, Shoemaker BA et al (2012) PubChem's BioAssay Database. *Nucleic Acids Res* 40:D400–D412
- Wang Y, Xiao J, Suzek TO, Zhang J, Wang J, Bryant SH (2009) PubChem: a public information system for analyzing bioactivities of small molecules. *Nucleic Acids Res* 37:W623–W633
- Boratyn GM, Camacho C, Cooper PS, Coulouris G, Fong A, Ma N, Madden TL, Matten WT, McGinnis SD, Merezuk Y et al (2013) BLAST: a more efficient report with usability improvements. *Nucleic Acids Res* 41:W29–W33
- Altschul SE, Gish W, Miller W, Myers EW, Lipman DJ (1990) Basic local alignment search tool. *J Mol Biol* 215:403–410
- Johnson M, Zaretskaya I, Raytselis Y, Merezuk Y, McGinnis S, Madden TL (2008) NCBI BLAST: a better web interface. *Nucleic Acids Res* 36:W5–W9
- Ye J, McGinnis S, Madden TL (2006) BLAST: improvements for better sequence analysis. *Nucleic Acids Res* 34:W6–W9
- Altschul SE, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* 25:3389–3402
- Morgulis A, Coulouris G, Raytselis Y, Madden TL, Agarwala R, Schaffer AA (2008) Database indexing for production MegaBLAST searches. *Bioinformatics* 24:1757–1764

Chapter 2

UniProtKB/Swiss-Prot, the Manually Annotated Section of the UniProt KnowledgeBase: How to Use the Entry View

Emmanuel Boutet, Damien Lieberherr, Michael Tognolli,
Michel Schneider, Parit Bansal, Alan J. Bridge, Sylvain Poux,
Lydie Bougueleret, and Ioannis Xenarios

Abstract

The Universal Protein Resource (UniProt, <http://www.uniprot.org>) consortium is an initiative of the SIB Swiss Institute of Bioinformatics (SIB), the European Bioinformatics Institute (EBI) and the Protein Information Resource (PIR) to provide the scientific community with a central resource for protein sequences and functional information. The UniProt consortium maintains the UniProt KnowledgeBase (UniProtKB), updated every 4 weeks, and several supplementary databases including the UniProt Reference Clusters (UniRef) and the UniProt Archive (UniParc).

The Swiss-Prot section of the UniProt KnowledgeBase (UniProtKB/Swiss-Prot) contains publicly available expertly manually annotated protein sequences obtained from a broad spectrum of organisms. Plant protein entries are produced in the frame of the Plant Proteome Annotation Program (PPAP), with an emphasis on characterized proteins of *Arabidopsis thaliana* and *Oryza sativa*. High level annotations provided by UniProtKB/Swiss-Prot are widely used to predict annotation of newly available proteins through automatic pipelines.

The purpose of this chapter is to present a guided tour of a UniProtKB/Swiss-Prot entry. We will also present some of the tools and databases that are linked to each entry.

Key words Swiss-Prot, TrEMBL, UniProt, Protein database, Amino-acid sequence, Manual annotation

1 Introduction

In late 2002 the SIB Swiss Institute of Bioinformatics (SIB), the European Bioinformatics Institute (EBI) and the Protein Information Resource (PIR) (*see Note 1*) joined forces by creating the Universal Protein Resource (UniProt) consortium [1]. The aim of this consortium is to provide high quality protein databases that are freely accessible to the scientific community.

The centerpiece of UniProt is the UniProt Knowledgebase (UniProtKB, <http://www.uniprot.org>), a comprehensive and

annotated protein sequence knowledgebase, which consists of two sections: UniProtKB/Swiss-Prot, containing manually expertly annotated entries, and UniProtKB/TrEMBL, containing computer translation and annotation of CoDing Sequences (CDS) extracted from the European Molecular Biology Laboratory nucleotide sequence database (EMBL) [2, 3] as well as sequences and annotation imported from Ensembl (<http://www.ensembl.org>), EnsemblGenomes (<http://ensemblgenomes.org>) including EnsemblPlants (<http://plants.ensembl.org>), and in the future, from RefSeq (<http://www.ncbi.nlm.nih.gov/refseq/>). Taking advantage of the expertly curated UniProtKB/Swiss-Prot section, automatic annotation procedures based on well described proteins are created and maintained to improve the annotation of related proteins in the UniProtKB/TrEMBL section. UniProtKB entries contain information curated by biologists or produced by annotation rules, and provide users with cross-links to about 140 external databases and give access to additional information or tools. UniProtKB/Swiss-Prot contributes actively to the “Gene Ontology” (GO, 10) annotation effort of proteins by manually assigning GO terms during the annotation process.

UniProtKB/Swiss-Prot is characterized by extended expert annotation (sequence properties, corresponding literature, etc.), minimal redundancy (separate entries for the same gene product in a given species and same cultivar/isolate are merged into a single protein entry), integration with other databases (cross-links to other life science databases including sequence-related databases as well as specialized data collections) and documentation (large number of index files and specialized documentation files) (*see Note 2*).

UniProtKB/TrEMBL, a computer-annotated database, mainly consists of translations of all coding sequences (CDS) proposed by the submitters to the EMBL/GenBank/DDJB nucleotide databases, which are not integrated into UniProtKB/Swiss-Prot, and by proteomes imported from Ensembl and EnsemblPlants. Some additional protein sequences are also extracted from the literature or directly submitted to UniProtKB. In addition to the preliminary information given by the submitters, UniProtKB/TrEMBL entries are processed according to automatic annotation procedures such as: (i) transfer of general annotation, domains and functional sites from well-characterized UniProtKB/Swiss-Prot entries belonging to protein family groups defined by InterPro [4], (ii) removal of redundancy by merging identical full-length sequences from the same organism, (iii) attribution of evidence to identify the source of individual data items (*see Note 3*).

In addition to UniProtKB, the UniProt consortium maintains several other protein databases, including:

- **The UniProt Archive (UniParc)**, which stores and maps all publicly available protein sequences from numerous databases, including UniProtKB, RefSeq, Patent offices, etc. (obsolete data excluded from UniProtKB are also present in UniParc)

- **The UniProt Reference Clusters (UniRef)**, which consists of clusters of sequences sharing 100 % identity for UniRef100, 90 % for UniRef90 and 50 % for UniRef50 (*see Note 4*). These databases are based on both UniProtKB and UniParc.

The Swiss-Prot group has initiated the Plant Proteome Annotation Program (PPAP) in 2001 [5] (<http://www.uniprot.org/program/plants/>). The current priority of this program is to annotate the proteomes of *Arabidopsis thaliana* and *Oryza sativa*, but without neglecting to annotate the proteins from other plant species. Our goals are the annotation of characterized plant specific and plant family proteins according to the Swiss-Prot standards [3]. At the beginning of March 2014 (UniProt release 2014_02), 34,824 plant sequence entries are present in UniProtKB/Swiss-Prot. Among them 12,665 are from *A. thaliana* and 3130 from *O. sativa*. In UniProtKB/Swiss-Prot, more than 1976 different plant species are present with at least one annotated protein (up-to-date statistics are available at <http://www.uniprot.org/statistics/>, <http://web.expasy.org/docs/relnotes/relnotes.html> and <http://www.uniprot.org/program/plants/statistics>).

To cope with the large and growing amount of sequenced genomes, UniProt assigns unique proteome identifiers giving the possibility to select proteins of a given organism. A subset of well-studied or biomedically and biotechnologically interesting organisms, selected to provide broad coverage of the tree of life, are manually defined as standard for a particular user community, and their proteome are “Reference proteomes” (*see Note 5*).

2 Materials

UniProtKB is hosted by uniprot.org (*see Note 6*). This chapter will always refer to the UniProtKB interface format used by the uniprot.org server (<http://www.uniprot.org/>), and will focus on UniProtKB/Swiss-Prot entries. The database is updated every four weeks. It is possible to download a local version of UniProtKB (*see Notes 7 and 8*).

2.1 UniProtKB Entries

2.1.1 Download and Display Content

The main distribution format of UniProtKB is a custom text-based format. Entries are represented by lines beginning with a two-letter code that identifies the type of data contained in the line. Each line follows a strictly defined format and the lines themselves are organized in such a way as to be easily legible to human users and simple to parse by computer programs (<http://www.expasy.org/sprot/userman.html#entrystruc>). However, UniProtKB proteins are also available in the more modern and structured XML/RDF format for computational use (<http://www.uniprot.org/docs/uniprot.xsd>).

2.1.2 Web View of an Entry

When accessing UniProtKB entries from the uniprot.org server, the default format is topic-wise organized in a user-friendly format when compared to the text-based format (*see* Fig. 1). The general elements of an entry in the uniprot.org view format are (from top to bottom): (i) UniProt header and search tool, (ii) UniProt tools (BLAST, alignment, mapping/retrieval in batch), (iii) general help, contact and basket tools, (iv) the header of the UniProtKB entry, (v) tools applicable to the current UniProtKB entry, (vi) current UniProtKB entry centric comment, feedback and external data tools, (vii) UniProtKB entry's section navigation bar organized by topics, (viii) the content of the current UniProtKB entry, (ix) details about the history of the current UniProtKB entry.

2.1.3 Content of an Entry

In most cases, each entry corresponds to a protein sequence encoded by a single gene locus (*see* **Note 9**). However, a few protein entries contain different coding loci merged into a single record when these loci are highly similar (e.g., histones, ubiquitins). References to residue positions within a sequence are made using sequential numbering starting with 1 at the N-terminal position. Displayed sequences correspond to the precursor forms of proteins, before posttranslational modifications and processing.

2.2 Tools and Databases Linked to UniProtKB

The uniprot.org website provides dedicated tools designed to exploit both protein sequences (BLAST, [6], alignments, database identifier mapping tool) and functional annotations (friendly but advanced search tool). SIB has developed the **Expert Protein Analysis System** proteomic server (ExPASy), which is another entry point to UniProtKB [7–9]. On <http://www.expasy.org/>, tools are available to deal with several aspects of protein analysis, including BLAST search, proteomics and sequence analysis, and take into account all splice variants as annotated in UniProtKB (*see* **Note 10**). Results obtained by these tools or links from other specific databases points to the corresponding UniProtKB entries.

3 Methods

3.1 Introduction

The main goal of UniProt is to provide a central resource for protein sequences and functional annotation. Together with UniProtKB/TrEMBL, UniProtKB/Swiss-Prot contains all known proteins, without species restriction. Currently the plant protein entries represent about 20 % of eukaryotes proteins and 7 % of the total content of UniProtKB/Swiss-Prot and our main effort is focused on the annotation of plant specific proteins characterized in literature from *Arabidopsis thaliana* and *Oryza sativa*. Any new genome fully sequenced, deposited in the public nucleotide database (EMBL/GenBank/DDBJ) and for which a gene prediction

UniProt UniProtKB

BLAST Align Upload lists

O80452
AMPD_ARATH
Reviewed (Swiss-Prot)

Protein AMP deaminase - *Arabidopsis thaliana* (Mouse-ear cress)
Protein Existence¹: Evidence at protein level

Gene AMPD, FAC1, At2g38280, F16M14.21

Display

Function¹
AMP deaminase plays a critical role in energy metabolism. Essential for the transition from zygote to embryo.

Catalytic activity
AMP + H₂O = IMP + NH₃.

Cofactor
Binds 1 zinc ion per subunit.

Enzyme regulation
Activated by ATP. Activated by sulfate ions (in vitro). Inhibited by phosphate ions.

Kinetics
K_M=6.7 mM for AMP (in the absence of ATP)
K_M=0.26 mM for AMP (in the presence of 1 mM ATP)
V_{max}=17 μmol/min/mg enzyme (in the absence of ATP)
V_{max}=375 μmol/min/mg enzyme (in the presence of 1 mM ATP)

Pathway
Purine metabolism; IMP biosynthesis via salvage pathway; IMP from AMP: step 1/1.

Sites

Feature key	Position(s)	Length	Description	Graphical view	Feature identifier
Metal binding ¹	391	1	Zinc; catalytic		
Metal binding ¹	393	1	Zinc; catalytic		
Binding site ¹	393	1	Substrate		
Metal binding ¹	659	1	Zinc; catalytic		
Binding site ¹	662	1	Substrate		
Active site ¹	681	1	Proton acceptor <input type="button" value="Inferred"/>		
Metal binding ¹	736	1	Zinc; catalytic		

Regions

Feature key	Position(s)	Length	Description	Graphical view	Feature identifier
Nucleotide binding ¹	289 - 296	8	ATP <input type="button" value="Reviewed Prediction"/>		

Entry version 93 (24 Jul 2013)
Sequence version 2 (01 Jun 2002)

Fig. 1 Header of a UniProtKB entry in the uniprot.org display format; partial view (<http://www.uniprot.org/uniprot/O80452>)

has been performed will be processed automatically. The predicted set of proteins is added to the UniProtKB/TrEMBL section as soon as the data is publicly available.

One of the great strengths of the UniProt Knowledgebase is the extensive integration and interconnectivity of numerous tools and external databases. The knowledgebase is cross-linked to about 140 other databases while most of the tools are adapted to allow analysis of all spliced isoforms described in the entry.

The UniProt Knowledgebase is constantly evolving and all recent modifications are detailed at http://www.uniprot.org/help/?query=* &fil=section:news while the forthcoming modifications are listed in <http://www.uniprot.org/changes>.

To further improve the quality of our annotation, we encourage users to submit comments and update requests ([http://www.uniprot.org/update?entry=primary accession number](http://www.uniprot.org/update?entry=primary%20accession%20number) accessible by the buttons and links present in each UniProtKB entries, *see* Fig. 1 iii and vi).

3.2 Accessing and Analyzing UniProtKB Entries

1. Quick and advanced text search (*see* Fig. 1 i) can be accessed directly from the UniProt home page (<http://www.uniprot.org>) (*see* Note 11). The advanced text search is designed to help users in writing complex queries by restricting terms to specific fields of the database (*see* Fig. 2 i), organized in the same topics of entry's sections. "Intelligent" filters are suggested to restrict the query with most likely terms (*see* Fig. 2 ii). Proteins of interest can be stored in the "basket" by checking boxes (*see* Figs. 2 iii and 3 i) and clicking on the button "Add to basket" for later comparison or download. When accessing the basket (*see* Fig. 3 ii), previously selected entries are listed and different actions are available: "Align", "BLAST", and "Download" (*see* Fig. 3 iii). The result table can be customized to fit user's requirement (*see* Fig. 4). A drag and drop

Entry	Entry name	Protein names	
<input type="checkbox"/>	Q01432	AMPD3_HUMAN	AMP deaminase 3
<input type="checkbox"/>	Q01433	AMPD2_HUMAN	AMP deaminase 2
<input type="checkbox"/>	P23109	AMPD1_HUMAN	AMP deaminase 1
<input type="checkbox"/>	Q90BTS	AMPD2_MOUSE	AMP deaminase 2
<input type="checkbox"/>	Q02356	AMPD2_RAT	AMP deaminase 2
<input type="checkbox"/>	O08739	AMPD3_MOUSE	AMP deaminase 3
<input type="checkbox"/>	P15274	AMPD_YEAST	AMP deaminase
<input type="checkbox"/>	Q540D0	AMPD_DICDI	AMP deaminase

Fig. 2 Text search result; partial view. Partial view of the result of a text search made on UniProtKB with "amp deaminase" as query

The screenshot shows the UniProt basket interface. At the top right, there is a 'Basket' icon with a count of 3. Below it, a modal window displays a table of three UniProtKB entries:

Entry	Entry name	Organism	Remove
P23109	AMPD1_HUMAN	Homo sapiens (Human)	
Q01433	AMPD2_HUMAN	Homo sapiens (Human)	
Q01432	AMPD3_HUMAN	Homo sapiens (Human)	

Below the table are buttons for 'Align', 'BLAST', 'Download', 'Clear', and 'Full View'. A 'Format:' dropdown menu is open, showing options: Text, FASTA (canonical), FASTA (canonical & isoform), Tab-delimited, Text (highlighted), Excel, GFF, XML, and RDF/XML. The background shows a partial view of the main UniProt table with entries like Q01432, Q01433, P23109, and Q9DBT5.

Fig. 3 The UniProt basket. View of the UniProt basket containing three UniProtKB protein entries (e.g., P23109, Q01433, and Q01432)

Customize results table

The screenshot shows the 'Customize results table' interface. It includes a 'Your columns' section with a drag-and-drop area containing 'Entry name', 'Protein names', 'Gene names', and 'Organism'. A 'Reset to default' button and 'Save'/'Cancel' buttons are present. Below this is the 'Additional columns' section, which has a search box containing 'e.g. gene, ontology,...'. A grid of category buttons is displayed, including:

- Names & Taxonomy
- Sequences
- Function
- Miscellaneous
- Interaction
- Structure
- Gene Ontology (GO)
- Expression (expanded)
- Subcellular location
- PTM / Processing
- Pathology & Biotech
- Developmental stage
- Date of
- Family & Domains
- Publications
- Induction (checked)
- Tissue specificity

At the bottom, there is a 'Databases' section with buttons for various data types like Sequence, 3D structure, Protein-protein, Chemistry, Protein family/group, PTM, Polymorphism, 2D gel, Proteomic, Protocols and materials, Genome annotation, Organism-specific, Phylogenomic, Enzyme and pathway, Other, and Gene expression.

Fig. 4 The UniProt customization interface. View of the UniProt customization tool

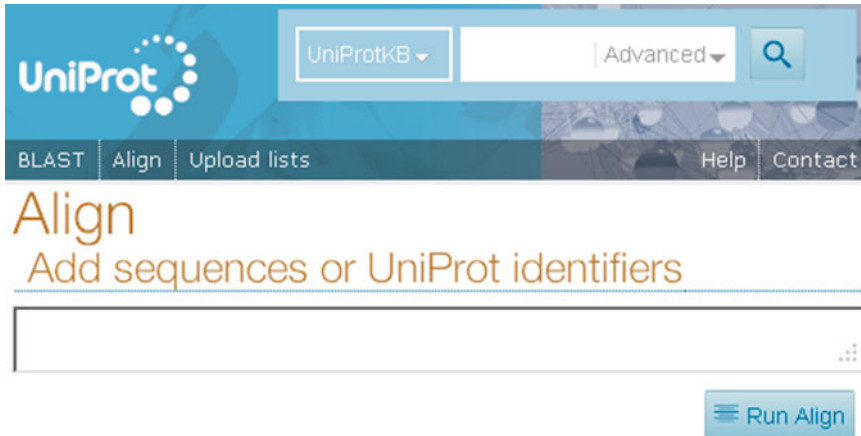


Fig. 5 The UniProt alignment tool. View of the UniProt protein alignment tool

tool makes it possible to change column order (*see* Fig. 4 i). A search engine is available to select for a favorite topic to display in the result table (*see* Fig. 4 ii). Each entry section can also be browsed in details (*see* Fig. 4 iii). When downloading selected entries in “tab-delimited” format, the columns of the output file are the same as the personalized display (*see* Fig. 2 iv). UniProt web services follow the representational state transfer (REST) architectural style to help sharing or storing favorite requests; this also permits easy programmatic access (*see* <http://www.uniprot.org/faq/28>).

2. An alignment tool based on Clustal Omega (<https://www.ebi.ac.uk/Tools/msa/clustalo/>) is available at <http://www.uniprot.org/align/> (*see* Fig. 5). The alignment output (*see* Fig. 6) is interactive and gives the possibility to highlight in different colors sequence features (*see* Fig. 6 ii) annotated in UniProtKB as well as amino acid properties by selecting properties of interest (*see* Fig. 6 i). When more than two protein sequences are aligned, an alignment tree is also available.
3. BLAST is available at <http://www.uniprot.org/blast/> (*see* Fig. 7). Standard parameters can be modified, default settings being: UniProtKB for the data set, 10 for the E-threshold, Matrix auto, no low complexity filtering and gap allowed (*see* **Note 12**). The BLAST output (*see* Fig. 8) gives, on the top, a list of sequences classified by level of similarity to the query, displayed in a graphical view of the query sequence with a similarity-dependent color gradient, and linked to the corresponding UniProtKB entries (*see* Fig. 8 i). A mechanism to allow the user to toggle between similarity based graphics and e-value based graphics will be soon available. All splice variants

Align
Display

AI

Download

-
- ALIGNMENT
-
-
- TREE
-
-
- RESULT INFO

Alignment

Highlight

Annotation

-
- Metal binding
-
-
- Alternative sequence
-
-
- Natural variant
-
-
- Chain
-
-
- Region
-
-
- Binding site
-
-
- Modified residue
-
-
- Sequence conflict
-
-
- Active site

Amino acid properties

-
- Similarity
-
-
- Hydrophobic
-
-
- Negative
-
-
- Positive
-
-
- Aliphatic
-
-
- Tiny
-
-
- Aromatic
-
-
- Charged
-
-
- Small
-
-
- Polar
-
-
- Big
-
-
- Serine Threonine

Accession	Protein Name	Length	Sequence	Score
P23109	AMPD1_HUMAN	1	-----MNVRFYFS	8
Q01433	AMPD2_HUMAN	1	MNRGQGLFRLSRCLFHSQSLPLGAGRRKGLDVAEPGPSRCRSDSPAVALAVVPAMASYP	60
Q01432	AMPD3_HUMAN	1	0
:				
P23109	AMPD1_HUMAN	260	DEPKPLYPNLDFTLDDMNFLLALIAQGPVKTYTHRRKLFSSKFQVHQMLNEMDELKEL	319
Q01433	AMPD2_HUMAN	342	CSEVELPYDPLQEFVADVNVLMALILINGPKSFCYRRQLYLSRFQMHVLLNEMKELAAQ	401
Q01432	AMPD3_HUMAN	241	QEPHSLPYDLETYTVDMSHILALITDGPTKTYCHRRLNFLSKFSLHEMLNEMSEFKEL	300
:				
P23109	AMPD1_HUMAN	320	KNNPHRDFYNCRKVDTHIAA...KQKHLRFIKKSYQIDADRVVYSTKEKNLTKELFA	379
Q01433	AMPD2_HUMAN	402	KKVPHRDFYNIKRVDTIHAS...DKHLLRFIKRANKHLEEIVHVEQGREQTLREVF	461
Q01432	AMPD3_HUMAN	301	KSNPHRDFYVNRKVDTHIAA...KQKHLRFIKHTYQTEPDRVTAEKGRKITLRQVDF	360
:				
P23109	AMPD1_HUMAN	380	KLKMPYDLTVDSLDVHAGRQTFQRFDFKFNKYNPVGASELRDLYLKTNDYINGEYFATI	439
Q01433	AMPD2_HUMAN	462	SNNLTAYDLSVDTLVDHADRNTFHRFDKFNKYNPVGASELRDLYLKTNDYINGEYFATI	521
Q01432	AMPD3_HUMAN	361	GLHMDPYDLTVDSLDVHAGRQTFHRFDKFNKYNPVGASELRDLYLKTNDYINGEYFARM	420
:				
P23109	AMPD1_HUMAN	440	IKEVGADLVEAKYQHAEPRLSIYGRSPDEWKLSSWFVNCNRHICPNMTHMIQVPRIDYF	499
Q01433	AMPD2_HUMAN	522	IKEVMSDLEESKYQNAELRLSIYGRSRDEWDLKARMAVHHRVHSPNVRVLVQVPRLFDVY	581
Q01432	AMPD3_HUMAN	421	VKEVARELEESKYQSEPRLSIYGRSPEEWNPLAYWFIQHKVYSPNMRWIIQVPRIDYF	480
:				
P23109	AMPD1_HUMAN	500	RSKNFLPHFGKMLNIFMPVFEATINPQADPELSVFLKHTGFSVDDESKEHSHMFSSK	559
Q01433	AMPD2_HUMAN	582	RTKGQLANFQEMLENIFLPLFEATVHPASHPDLHLFLEHVDFGFSVDDESKPENHVNLE	641
Q01432	AMPD3_HUMAN	481	RSKLLLPNFGKMLNIFLPLFKATINPQDREHLHLFLKYVTGFSVDDESKEHSHMFSSK	540
:				
P23109	AMPD1_HUMAN	560	SPKQEWLEKNPSTYTYAYMYANIIVLNSLRKERGMNTFLFRPHCGEAGALTHLMTAF	619
Q01433	AMPD2_HUMAN	642	SPLPEANVVEENPPYAYLYYTFANMAMNLHRORQGFHTFVLRPHCGEAGAIHHLVSFAF	701
Q01432	AMPD3_HUMAN	541	SPNPDVUTSEQNPPTSYLYMYANIIVLNLRRERGLSTFLFRPHCGEAGSITHLVSFAF	600
:				
P23109	AMPD1_HUMAN	620	MIADDISHGLNLKSPVLQYLFLAQIP IAMSPLSMNSLFLEYAKNPFLDFLQKGLNISL	679
Q01433	AMPD2_HUMAN	702	HLAENISHGLLLKAPVLQYLFLAQIGIAMSPLSMNSLFLEYAKNPFLDFLQKGLNISL	761
Q01432	AMPD3_HUMAN	601	LTADNISHGLLLKSPVLQYLFLAQIP IAMSPLSMNSLFLEYAKNPFLDFLQKGLNISL	660
:				
P23109	AMPD1_HUMAN	680	STDPPHQFHTKEPLHEEYAIQAQVFKLSTCDNCEVARNVSLVQCGGISHEEKVKFLGDNYL	739
Q01433	AMPD2_HUMAN	762	STDPLQFHFHTKEPLHEEYAIATQVWKLSSCDNCEVARNVSLVQCGGISHEEKVKFLGDNYL	821
Q01432	AMPD3_HUMAN	661	STDPPHQFHYTKALHEEYAIQAQVWKLSTCDLCEIARNVSLVQCGGLSHQEKQKFLGQNY	720

Fig. 6 Protein alignment result. Partial view of the protein alignment result made on UniProtKB for P23109, Q01433 and Q01432 protein entries

annotated in UniProtKB are considered during the BLAST (their UniProt accessions are followed by “-n” where “n” is a digit for Swiss-Prot alternative splicing products) (see Fig. 8 i). On the lower part of the output BLAST result, a detailed list of the matched proteins is displayed, with a graphical view of the best alignment for each hit represented in a graphical view with the color code described previously, and linked to all corresponding local alignments between the query and the hit sequences (see Fig. 8 ii). All options available for text search result are applicable to this list (see Fig. 4).

- Database Entries can be downloaded in batch. Several sets of protein sequences are proposed for download at <http://www.uniprot.org/downloads>. Entries present in the basket can be retrieved in different formats (see Fig. 3 iv). A dedicated tool to convert and download a list of proteins is available at

Add sequence or UniProt identifier

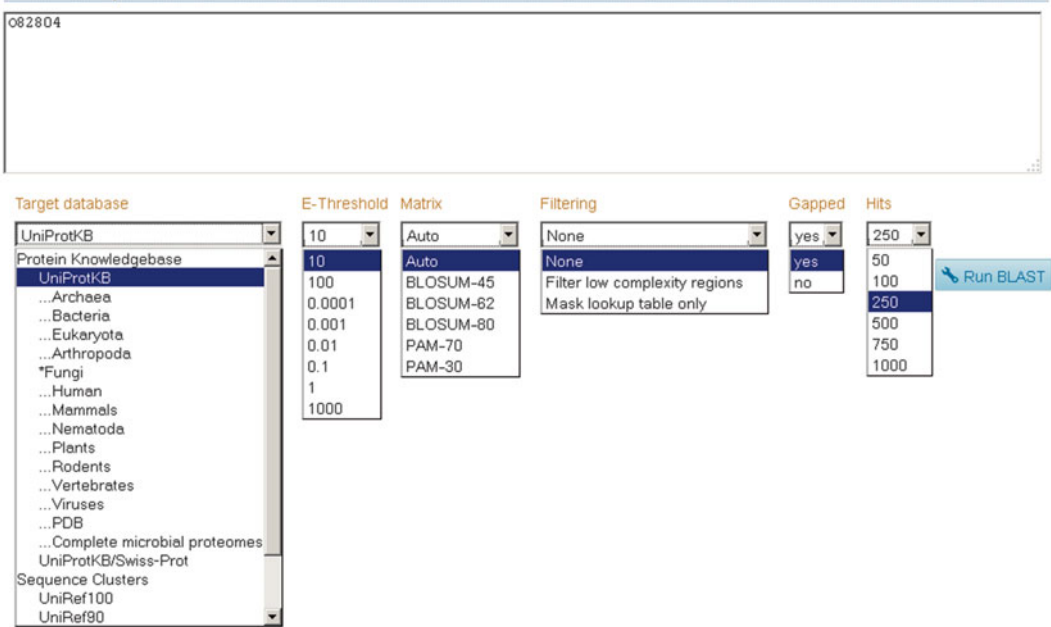


Fig. 7 The UniProtBLAST tool. View of the UniProt BLAST tool



Fig. 8 BLAST result. Partial view of the result of the BLAST made on UniProtKB with O82804 entry as query

Upload Lists

1. Provide your identifiers

e.g. P31946 P62258 ALBU_HUMAN EFTU_ECOLI

OR upload your own file: No file selected.

2. Select options

The screenshot shows the UniProt upload tool interface. At the top, there is a text input field containing the example identifiers: "e.g. P31946 P62258 ALBU_HUMAN EFTU_ECOLI". Below this, there is a button labeled "Browse..." and the text "No file selected.". The main part of the interface is divided into two columns: "From" and "To".

The "From" column has a dropdown menu labeled "i" with "UniProtKB AC/ID" selected. The list of options includes: UniProt, UniProtKB AC/ID, UniParc, UniRef50, UniRef90, UniRef100, Other sequence databases (EMBL/GenBank/DBJ, EMBL/GenBank/DBJ CDS, PIR, UniGene, Entrez Gene (GeneID), GI number*, IPI, RefSeq Protein, RefSeq Nucleotide), and 3D structure databases (PDB, DisProt, HSSP).

The "To" column has a dropdown menu labeled "ii" with "UniProtKB" selected. The list of options includes: 3D structure databases (PDB, DisProt, HSSP), Protein-protein interaction databases (DIP, MINT), Protein family/group databases (Allergome, MEROPS, PeroxiBase, PptaseDB, REBASE, TCDB), Polymorphism databases (DMDM), and 2D gel databases (Aarhus/Ghent-2DPAGE, ECO2DBASE, World-2DPAGE). A "Go" button is located at the top right of the "To" column.

Fig. 9 The UniProt downloading tool. View of the UniProt downloading tool

www.uniprot.org/uploadlists/ (see Fig. 9). The user provides a list of accessions in any of the supported formats (see <http://www.uniprot.org/help/uploadlists> and Fig. 9 i) and can convert this list into any of the listed databases (see Fig. 9 ii). When the “from” database is “UniProtKB (AC/ID)” and the “to” database is “UniProt”, the user can retrieve UniProtKB protein entries from a UniProtKB accession list.

- UniProtKB entries are also present or cross-linked in several other biological databases and tools such as ExpASY (<http://www.expasy.org/>), the NCBI (<http://www.ncbi.nlm.nih.gov/protein/>) and TAIR (<http://www.arabidopsis.org>).

3.3 The Web View of a UniProtKB Entry

3.3.1 UniProt Banner

When accessing the UniProt website, some elements are always present at the top of the page: the UniProt logo to return to the home page, the search box (see Fig. 1-i), access to additional tools including BLAST, alignment and download, described elsewhere (see Fig. 1-ii), links to help (see **Note 13**), contact, and to the basket containing selected entries (see Fig. 1-iii).

- 3.3.2 Entry Header** The first block of each entry details (*see* Fig. 1-iv) accession numbers, status (*reviewed* for UniProtKB/Swiss-Prot and *unreviewed* for UniProtKB/TrEMBL), as well as protein and gene names and synonyms. The primary accession number (AC, e.g., O80452) of an entry (*see* **Note 14**, documentation available at http://www.uniprot.org/manual/accession_numbers) is stable and provides a unique identifier which allows unambiguous citation of the entry (*see* **Notes 15** and **16**). The entry name (ID, e.g., AMPD_ARATH) consists of up to 11 characters and takes the general form X_Y. Both X and Y represent mnemonic codes of up to 5 alphanumeric characters for both the protein name (X) and the species (Y) (documentation is available at http://www.uniprot.org/manual/entry_name). Entry names, corresponding to protein/gene name abbreviations, are subject to revision and therefore do not provide a stable means of identifying individual entries. Because entry names are prone to change, researchers who wish to cite entries in publications should always cite the primary accession number.
- 3.3.3 Analysis Tabs** Direct access to BLAST, alignment, and download, tools described in Subheading 3.2, is available from the protein entry view (*see* Fig. 1-v). Entry can also be stored in the basket.
- 3.3.4 Contribution to Entry Annotation Tabs** Suggestions to update the content of the current entry can be sent via “comment” or “feedback” features (*see* Fig. 1-vi).
- 3.3.5 Entry's Section Navigation Panel** The content of a protein entry is organized in 15 topics. To navigate and switch between topics, a display menu containing direct links to the different blocks of the entry is always visible on the left side of the screen (*see* Fig. 1-vii). Check boxes in this menu permit to hide/display the corresponding section.
- 3.3.6 Entry Content View** In the main central area, the content of the current protein entry is displayed by thematic topics (*see* Fig. 1-viii). When a term is followed by “i” as exponent, this means that contextual information are available for this term.
- Most of the information in this section is extracted from the literature. Some information is also based on unproven empirical biological evidence, determined by computer prediction, or propagated from homologous members of the family (for details about annotation procedures, *see* <http://www.uniprot.org/faq/45>). In these cases, non-experimental qualifiers are added (*see* http://www.uniprot.org/manual/non_experimental_qualifiers); the qualifiers are: “Potential” for computer predicted, logical or conclusive evidence (*see* **Note 17**, represented on the website as “Reviewed prediction” in Swiss-Prot and as “Predicted” in TrEMBL), “Probable” for non-direct experimental evidence (*see* **Note 18**,

represented on the website as “Inferred” in Swiss-Prot), and “By similarity” for experimental evidence in a close member of the family. Explanations of non-experimental qualifiers can be obtained by clicking on them in the entry.

Annotations are mainly distributed in four different types:

1. *General annotation*: provides general information about the protein, mostly biological knowledge, in different subsections (*see* http://www.uniprot.org/manual/general_annotation).
2. *Sequence feature*: information associated with specific residues of the current protein sequence (*see* http://www.uniprot.org/manual/sequence_annotation). Each sequence feature contains a “Feature key” (*see* **Note 19**), “Position(s)” indicates limits of the feature according to the amino acid residue positions of the displayed sequence (*see* **Note 20**), the “Length” of the feature is also given, a “Description” of the feature (*see* **Note 21**), a “Graphical view” to visualize the region in the consensus sequence, and, when available, “Feature identifier”. UniProtKB/Swiss-Prot entries contain extensive annotation of all features that are predicted (and compatible with the protein function), experimentally proven, or determined by resolution of the protein structure.
3. *Cross-references*: used to point to information related to entries and found in data collections other than UniProtKB (*see* http://www.uniprot.org/help/cross_references_section).
4. *Ontologies and controlled vocabularies*: a combination of controlled vocabularies and ontologies is used to summarize the functional implication of the current protein. The controlled vocabulary is developed by UniProtKB/Swiss-Prot (*see* <http://www.uniprot.org/keywords/>), and GO terms (GO, [10]), a formal representation of terms that can be used to describe biological function, process and component, are developed and curated by the GO consortium (*see* http://www.uniprot.org/help/gene_ontology). Some keywords are derived from automatic annotation in UniProtKB/TrEMBL entries, but the vast majority is added manually in UniProtKB/Swiss-Prot entries. They describe the main characteristics of the protein.

The information contained in the entry is organized in a total of 15 topics, each accessible from the display panel. Depending on the information available in each entry, some sections might appear or not.

The 15 sections used in UniProtKB and their respective subsections are listed below:

1. “Function”: (*see* Fig. 10 and http://www.uniprot.org/help/function_section). Contains information pertinent to biological knowledge of the protein function.

Function

AMP deaminase plays a critical role in energy metabolism. Essential for the transition from zygote to embryo. [1 Publication](#)

Catalytic activity

AMP + H₂O = IMP + NH₃. [1 Publication](#)

Cofactor

Binds 1 zinc ion per subunit. [1 Publication](#)

Enzyme regulation

Activated by ATP. Activated by sulfate ions (in vitro). Inhibited by phosphate ions. [1 Publication](#)

Kinetics

K_M=6.7 mM for AMP (in the absence of ATP) [1 Publication](#)

K_M=0.26 mM for AMP (in the presence of 1 mM ATP)

V_{max}=17 μmol/min/mg enzyme (in the absence of ATP)

V_{max}=375 μmol/min/mg enzyme (in the presence of 1 mM ATP)

Pathway

Purine metabolism; IMP biosynthesis via salvage pathway; IMP from AMP: step 1/1.

Sites

Feature key	Position(s)	Length	Description	Graphical view	Feature identifier
Metal binding ⁱ	391	1	Zinc; catalytic		
Metal binding ⁱ	393	1	Zinc; catalytic		
Binding site ⁱ	393	1	Substrate		
Metal binding ⁱ	659	1	Zinc; catalytic		
Binding site ⁱ	662	1	Substrate		
Active site ⁱ	681	1	Proton acceptor Inferred		
Metal binding ⁱ	736	1	Zinc; catalytic		

Manual assertion based on experiment described in:

"Membrane association, mechanism of action, and structure of Arabidopsis embryonic factor 1 (FAC1)."
 Han B.W., Bingman C.A., Mahnke D.K., Bannen R.M., Bednarek S.Y., Sabina R.L., Phillips G.N. Jr
 J. Biol. Chem. 281:14939-14947(2006) [PubMed] [Europe PMC] [Abstract]
Cited for: X-RAY CRYSTALLOGRAPHY (3.3 ANGSTROMS) OF 140-839 IN COMPLEX WITH COFORMYCIN 5'-PHOSPHATE;

Regions

Feature key	Position(s)	Length	Description	Graphical view	Feature identifier
Nucleotide binding ⁱ	289 - 296	8	ATP Reviewed Prediction		

GO - Molecular functionⁱ

[AMP deaminase activity](#)

Inferred from genetic interactionⁱ Ref.1 Source: TAIR

[ATP binding](#)

Inferred from electronic annotationⁱ Source: UniProtKB-KW

[metal ion binding](#)

Inferred from electronic annotationⁱ Source: UniProtKB-KW

[Complete GO annotation...](#)

Keywords - Molecular functionⁱ

[Hydrolase](#)

Keywords - Biological processⁱ

[Nucleotide metabolism](#)

Keywords - Ligandⁱ

[ATP-binding](#), [Metal-binding](#), [Nucleotide-binding](#), [Zinc](#)

Enzyme and pathway databases

SABIO-RK	O80452 .
UniPathway	UPA00591 ; UER00663 .

Fig. 10 Function section of a UniProtKB entry. View of the “function” section of the UniProtKB protein O80452

The different subsections of the function section are:

- (a) General annotation dealing with function, catalytic activity, cofactor, enzyme regulation, biophysicochemical properties, and pathway
 - (b) Sequence features describing active site, metal binding, binding site, site, calcium binding, zinc finger, and DNA binding with a graphical view
 - (c) GO terms of the ‘Molecular function’ section
 - (d) Keywords of ‘Molecular function’, ‘Biological process’, and ‘Ligand’ subsections
 - (e) Cross-references that point to family, enzyme, and pathway databases
2. “Names & Taxonomy”: (*see* Fig. 11 and http://www.uniprot.org/help/names_and_taxonomy_section).
- This block describes protein names, gene names and taxonomy of the organism. The recommended protein name is given in the first row, followed by the alternative names used in the literature. In the case of an enzyme, the Enzyme Commission (EC) number is given as synonym. This EC number is an active link to the Enzyme database (<http://www.expasy.org/enzyme/>) [11], which contains detailed information about enzyme activity and lists all UniProtKB/Swiss-Prot entries having the same EC number. The second row of this block describes the gene encoding the protein in the following order: gene name, synonyms, ordered locus name when applicable (*see* Note 22) and ORF names used by the genomic sequencing projects, when available. Following the gene description, the organism name, the NCBI taxonomy identifier, and the summarized taxonomic hierarchy are actively linked to the UniProt taxonomy browser (<http://www.uniprot.org/taxonomy/>) which contains details on the organism and gives access to all UniProtKB entries of that organism (*see* Note 23).
3. “Subcellular location”: (*see* Fig. 12 and http://www.uniprot.org/help/subcellular_location_section).
- Contains information pertinent to biological knowledge of the protein localization and topology.
- The different subsections of the subcellular location section are:
- (a) General annotation dealing with subcellular location
 - (b) Sequence features describing transmembrane and topological domain with a graphical view
 - (c) GO terms of the ‘Cellular component’ section
 - (d) Keywords of the ‘Cellular component’ section
4. “Pathology & Biotech”: (*see* Fig. 13 and http://www.uniprot.org/help/pathology_and_biotech_section).

Names & Taxonomy

Protein names ⁱ	<p>Recommended name:</p> <p>AMP deaminase</p> <ul style="list-style-type: none"> ▪ Short name: AtAMPD ▪ EC: 3.5.4.6 <p>Alternative name(s):</p> <ul style="list-style-type: none"> • Protein EMBRYONIC FACTOR 1
Gene names ⁱ	<p>Name: AMPD</p> <p>Synonyms: FAC1</p> <p>Ordered Locus Names: At2g38280</p> <p>ORF Names: F16M14.21</p>
Organism ⁱ	Arabidopsis thaliana (Mouse-ear cress) [Reference proteome]
Taxonomic identifier ⁱ	3702 [NCBI]
Taxonomic lineage ⁱ	Eukaryota > Viridiplantae > Streptophyta > Embryophyta > Tracheophyta > Spermatophyta > Magnoliophyta > eudicotyledons > core eudicotyledons > rosids > malvids > Brassicales > Brassicaceae > Camelineae > Arabidopsis

Organism-specific databases

TAIR	AT2G38280.
------	------------

Fig. 11 Names & taxonomy section of a UniProtKB entry. View of the “names and taxonomy” section of the UniProtKB protein O80452

Subcellular location

Membrane; Single-pass membrane protein. Microsome membrane

Note: Might be associated with the inner mitochondrial membrane [By similarity](#) · [1 Publication](#) ▼

Topology

Feature key	Position(s)	Length	Description	Graphical view	Feature identifier
Transmembrane ⁱ	8 – 28	21	Helical; Reviewed Prediction		

GO - Cellular componentⁱ

cytosol

Inferred from direct assayⁱ Ref.7 [PubMed 21166475](#) Source: TAIR

endoplasmic reticulum

Inferred from electronic annotationⁱ Source: UniProtKB-KW

integral to mitochondrial outer membrane

Inferred from direct assayⁱ [PubMed 21896887](#) Source: TAIR

nucleus

Inferred from direct assayⁱ Ref.7 Source: TAIR

[Complete GO annotation...](#)

Keywords - Cellular componentⁱ

Endoplasmic reticulum, Membrane, Microsome

Fig. 12 Subcellular location section of a UniProtKB entry. View of the “subcellular location” section of the UniProtKB protein O80452

Pathology & Biotechⁱ

Allergenic properties

Causes an allergic reaction in human. Common symptoms of mite allergy are bronchial asthma, allergic rhinitis and conjunctivitis. Binds to IgE in 80% of patients with house dust allergy.

Mutagenesis

Feature key	Position(s)	Length	Description	Graphical view	Feature identifier
Mutagenesis ⁱ	132	1	C → A: Loss of activity.		
Mutagenesis ⁱ	150	1	N → E: Loss of N-glycosylation.		

Keywords - Diseaseⁱ

Allergen

Protein family/group databases

Allergome	1232. Der p 1.0111. 310. Der p 1.
-----------	--------------------------------------

Fig. 13 Pathology & biotech section of a UniProtKB entry. View of the “pathology and biotech” section of the UniProtKB protein P08176

Contains information pertinent to biological knowledge of disease(s) and phenotype(s) associated with the deficiency of the protein.

The different subsections of the Pathology & Biotech section are:

- (a) General annotation dealing with involvement in disease, natural variant, allergenic properties, biotechnological use, toxic dose, and pharmaceutical use
 - (b) Sequence features describing disruption phenotype and mutagenesis with a graphical view
 - (c) Keywords of the ‘Disease’ section
 - (d) Cross-references that point to organism-specific databases
5. “Post translational modification (PTMs) / Processing”:^{(see Fig. 14 and http://www.uniprot.org/help/ptm_processing_section)}





Contains information pertinent to biological knowledge of the protein posttranslational modifications.

The different subsections of the PTM / processing section are:

- (a) Sequence features describing initiator methionine, signal, pro-peptide, transit peptide, chain, peptide, modified residue, lipidation, glycosylation, disulfide bond, and cross-link with a graphical view
- (b) General annotation dealing with posttranslational modification

PTM / Processingⁱ

Molecule processing

Feature key	Position(s)	Length	Description	Graphical view	Feature identifier
Transit peptide ⁱ	1 – 55	55	Chloroplast 		
Chain ⁱ	56 – 333	278	Adenylate isopentenyltransferase 3, chloroplastic		PRO_0000391072
Propeptide ⁱ	334 – 336	3	Removed in mature form		PRO_0000396781

Amino acid modifications

Feature key	Position(s)	Length	Description	Graphical view	Feature identifier
Modified residue ⁱ	333	1	Cysteine methyl ester		
Lipidation ⁱ	333	1	S-farnesyl cysteine 		

Post-translational modificationⁱ

Farnesylated.

Keywords - PTMⁱ

Lipoprotein, Methylation, Prenylation

Proteomic databases

PRIDE ⁱ	Q93WC9.
--------------------	---------

Fig. 14 PTM/processing section of a UniProtKB entry. View of the “PTM/processing” section of the UniProtKB protein Q93WC9

- (c) Keywords of the ‘PTM’ section
- (d) Cross-references that point to proteomics and PTM databases
6. “Expression”: (*see* Fig. 15 and http://www.uniprot.org/help/expression_section).
Contains information pertinent to biological knowledge of the protein expression.
The different subsections of the expression section are:
 - (a) General annotation dealing with tissue specificity, developmental stage and induction
 - (b) Keywords of the ‘Developmental stage’ section
 - (c) Cross-references that point to gene expression databases
7. “Interaction”: (*see* Fig. 16 and http://www.uniprot.org/help/interaction_section).

Expression

Tissue specificity

Expressed in seedlings, roots, leaves, flowers, pollen grains, pollen tubes and siliques, and at a lower level in stems.

1 Publication

Developmental stage

Expressed in both male and female gametophytes, at the zygote stage, in the endosperm, and during early embryo development. Observed in cotyledonary embryos and in the basal part of the embryo, but not in the suspensor or in mature embryos. Also expressed during somatic embryogenesis.

1 Publication

Gene expression databases

Genevestigator	O80452.
----------------	---------

Fig. 15 Expression section of a UniProtKB entry. View of the “expression” section of the UniProtKB protein O80452

Interaction

Subunit structure

Homodimer. Interacts with AHK4. 2 Publications

Binary interactions

With	Entry	#Exp.	IntAct	Notes
AHK4	Q9C5U0	2	EBI-1807679,EBI-1100775	

Protein-protein interaction databases

IntAct	O80452. 2 interactions.
--------	-------------------------

Fig. 16 Interaction section of a UniProtKB entry. View of the “interaction” section of the UniProtKB protein O80452

Contains information pertinent to biological knowledge of the protein interactions.

The different subsections of the interaction section are:

- (a) General annotation dealing with subunit structure
 - (b) Specific annotation describing binary interactions
 - (c) Cross-references that point to protein–protein interaction databases
8. “Structure”: (*see* Fig. 17 and http://www.uniprot.org/help/structure_section).
- Contains information pertinent to biological knowledge of the protein structure.
- The different subsections of the structure section are:
- (a) Sequence features describing turn, beta strand and helix with a graphical view (when available)
 - (b) Cross-references that point to 3D structure databases
9. “Family & Domains”: (*see* Fig. 18 and http://www.uniprot.org/help/family_and_domains_section).



Fig. 17 Structure section of a UniProtKB entry. View of the “structure” section of the UniProtKB protein O80452

Contains information pertinent to biological knowledge of the protein family and domains

The different subsections of the Family & Domains section are:

- Sequence features describing domain, repeat, compositional bias, region, coiled coil, motif, and domain with a graphical view with a graphical view
 - General annotation dealing with sequence similarities; a comment describing to which family the protein may belong may be included. It is linked to a UniProt query that lists all UniProtKB entries belonging to the same family (*see* **Note 24** and Fig. 18 i). In the case of transporter families, the transport classification (TC) number is present when available, and a cross-link to the transport classification database (<http://www.tcdb.org>) is also included.
 - Keywords of the ‘Domain’ section
 - Cross-references that point to phylogenomic and family and domain databases
10. “Sequence”: (*see* Fig. 19 and http://www.uniprot.org/help/sequences_section).

Contains general metadata determined for the given sequence, such as sequence length, molecular weight, and CRC64 checksum (64 bit Cyclic Redundancy Check value) [12] (*see* **Note 25**). Each subsection contains information pertinent to biological knowledge of the protein sequence. On the right side of all sequences, a quick access to the FASTA format (http://en.wikipedia.org/wiki/FASTA_format) of the sequence and to sequence/proteomic tools is present (*see* Fig. 19 i).

Family & Domainsⁱ

Region

Feature key	Position(s)	Length	Description	Graphical view	Feature identifier
Region ⁱ	462 – 467	6	Substrate binding		
Region ⁱ	737 – 740	4	Substrate binding		

Compositional bias

Feature key	Position(s)	Length	Description	Graphical view	Feature identifier
Compositional bias ⁱ	86 – 92	7	Poly-Gly		
Compositional bias ⁱ	158 – 161	4	Poly-Asp		

Sequence similarities

Belongs to the [adenosine and AMP deaminases family](#).



Keywords - Domainⁱ

[Transmembrane](#), [Transmembrane helix](#)

Phylogenomic databases

eggNOG	COG1816 .
HOGENOM	HOG000092200 .
InParanoid	O80452 .
KO	K01490 .
OMA	HRVYSDN .
PhylomeDB	O80452 .
ProtClustDB	PLN02768 .

Family and domain databases

InterPro	IPR006650 . A/AMP_deam_AS. IPR001365 . A/AMP_deaminase_dom. IPR006329 . AMP_deaminase. [Graphical view]
PANTHER	PTHR11359 . PTHR11359. 1 hit.
Pfam	PF00962 . A_deaminase. 1 hit. [Graphical view]
TIGRFAMs	TIGR01429 . AMP_deaminase. 1 hit.
PROSITE	PS00485 . A_DEAMINASE. 1 hit. [Graphical view]

Fig. 18 Family & domains section of a UniProtKB entry. View of the “family and domains” section of the UniProtKB protein O80452

The different subsections of the sequence section are:

- The sequence status, either complete or fragment(s)
- Sequence processing when accurate; details about this processing are described in the “PTM/Processing” section
- The canonical protein sequence
- Alternative products with sequence and additional related information, when existing. The alternative products subsection describes the proteins which may be produced by

Sequence

Sequence status[†]: Complete.

O80452-1 [UniParc] [FASTA](#)

```

MEPNIIQLAL AALFGASFVA VSGPFMFHKA LMLVLERGKE RKENPDGDEP 50
QNPTLVRRRS QVRRKVNQY GRSPASLPDA TPFTDGGGGG GGDTRSNHG 100
YYVDEIPPGI PRLHTPSEGR ASVHGASSIR KTGSFVRPIS PKSPVASASA 150
...
EYSIAASVWK LSACDLCEIA RNSVIQSGFS HALKSHWIGK DYYKRGPDGN 800
DIHKTNPVPH RVEFRDTIWK EEMQQVYLKG AVISDEVVP 839
    
```

Length: 839
 Mass (Da): 95,130
 Last modified: June 1, 2002. Version 2.
 Checksum : 188F1F4A589A17DA[‡]

Blast

- Blast
- ProtParam
- Compute pI/MW
- ProtScale
- PeptideMass
- PeptideCutter

◀ Hide

Sequence caution
 The sequence [BAD94943.1](#) differs from that shown. Reason: Intron retention.

Sequence databases

Select the link destinations:	<input checked="" type="radio"/> EMBL <input type="radio"/> GenBank <input type="radio"/> DDBJ
	AC003028 Genomic DNA. Translation: AAC27176.2 . CP002685 Genomic DNA. Translation: AEC09516.1 . CP002685 Genomic DNA. Translation: AEC09517.1 . AY056301 mRNA. Translation: AAL07150.1 . AY133852 mRNA. Translation: AAM91786.1 . AK316943 mRNA. Translation: BAH19646.1 . AK221552 mRNA. Translation: BAD94943.1 . Sequence problems.
IPI	IPI00546126 .
PIR	T01259 .
RefSeq	NP_565886.1 . NM_129384.2 . NP_850294.1 . NM_179963.2 .
UniGene	At.12466 .

Genome annotation databases

EnsemblPlants	AT2G38280.1 ; AT2G38280.1 ; AT2G38280.1 . AT2G38280.2 ; AT2G38280.2 ; AT2G38280.2 .
GeneID	818408 .
KEGG	ath:AT2G38280 .

Fig. 19 Sequence section of a UniProtKB entry. View of the “sequence” section of the UniProtKB protein O80452

alternative splicing or promoter usage. Modifications of the canonical sequence necessary to produce the alternative product sequence are described in the sequence features subsection (see Fig. 20).

- (e) General annotation dealing with sequence caution, caution, polymorphism, RNA editing and mass spectrometry
- (f) Sequence features describing natural variant, alternative sequence, sequence uncertainty, sequence conflict, non-adjacent residues, non-terminal residue, and non-standard residue with a graphical view
- (g) Keywords of the ‘Coding sequence diversity’ section
- (h) Cross-references that point to sequence, genome annotation databases and polymorphism databases

Sequences (2)¹Sequence status¹: Complete.This entry describes 2 isoforms¹ produced by **alternative splicing**. **Isoform 1** (identifier: **O82804-1**) [UniParc] 

This isoform has been chosen as the 'canonical' sequence. All positional information in this entry refers to it. This is also the sequence that appears in the downloadable versions of the entry.

[Show »](#)

Length: 695
Mass (Da): 77,206
Last modified: November 1, 1998
 - v1

Checksum:
 607A0720ED381C08¹



Isoform 2 (identifier: **O82804-2**) [UniParc] 

The sequence of this isoform differs from the canonical sequence as follows:

339-339: N → K
 340-695: Missing.

[« Hide](#)

```
MKRGKDEEKI LEPMPFRLHV NDADKGGPRA PPRNKMALYE QLSIPSORFG 50
DHGTMNSRSN NTSTLVHGPV SSQPCGVERN LSVQHLDSSA ANQATEKFVS 100
QMSFMENVRS SAQHDQRKMV REEEDFAVPV YINSRRSQSH GRKSGIEKE 150
KHTPMVAPSS HHSIRFQEVN QTGSKQNVCL ATCSKPEVRD QVKANARSGG 200
FVISLDVSVT EEIDLEKSAS SHDRVNDYNA SLRQESRNL YRDGGKTRLK 250
DTDNGAESH L ATENHSQEGH GSPEDIDNDR EYSKSPACAS LQQINEEASD 300
DVSDDSMVDS ISSIDVSPDD VVGILGQKRF WRARKAIAK 339
```

Length: 339
Mass (Da): 37,760

Checksum:
 4CBEAD87D3292DA6¹



Note: No experimental confirmation available.**Sequence conflict**The sequence [CAA72719.1](#) differs from that shown. Reason: Frameshift at positions 437, 472 and 485.**Caution¹**

Isoform-2 : No experimental confirmation available.

Alternative sequence


Feature key	Position(s)	Length	Description	Graphical view	Feature identifier
Alternative sequence ¹	339	1	N → K in isoform 2.		VSP_004042
Alternative sequence ¹	340 - 695	356	Missing in isoform 2.		VSP_004043

Fig. 20 Sequence section of a UniProtKB entry containing alternative products. View of the “sequence” section of the UniProtKB protein O82804; only details concerning the alternative splicing are shown

11. “Cross-references”: (*see* Fig. 21 and http://www.uniprot.org/help/cross_references_section).

The cross-references section is divided into subsections organized by themes. This section links the protein to several other databases that contain information relevant to that protein. Many of these cross-links are automatically added to UniProtKB/TrEMBL entries, but some are manually created in UniProtKB/Swiss-Prot entries (*see* **Note 19**). Each row of this block corresponds to a single database, the name of which

Sequence databases

Select the link destinations:	AC003028 Genomic DNA. Translation: AAC27176.2 . CP002685 Genomic DNA. Translation: AEC09516.1 . CP002685 Genomic DNA. Translation: AEC09517.1 . AY056301 mRNA. Translation: AAL07150.1 . AY133652 mRNA. Translation: AAM91786.1 . AK316943 mRNA. Translation: BAH19646.1 . AK221552 mRNA. Translation: BAD94943.1 . Sequence problems.
<input checked="" type="radio"/> EMBL	
<input type="radio"/> GenBank	
<input type="radio"/> DDBJ	
	IPI IP100546126.
	PIR T01259.
	RefSeq NP_565886.1. NM_129364.2. NP_850294.1. NM_179963.2.
	UniGene At.12466.

3D structure databases

Select the link destinations:	Entry	Method	Resolution (Å)	Chain	Positions	PDbsum
<input checked="" type="radio"/> PDBe	2A3L	X-ray	3.34	A	140-839	[*]
<input type="radio"/> RCSB PDB						
<input type="radio"/> PDBj						
ProteinModelPortal	O80452.					
SMR	O80452. Positions 212-839.					
ModBase	Search...					

Protein-protein interaction databases

IntAct	O80452. 2 interactions.
--------	-------------------------

Proteomic databases

PaxDb	O80452.
PRIDE	O80452.

Protocols and materials databases

StructuralBiologyKnowledgebase	Search...
--------------------------------	-----------

Genome annotation databases

EnsemblPlants	AT2G38280.1 ; AT2G38280.1 ; AT2G38280 . AT2G38280.2 ; AT2G38280.2 ; AT2G38280 .
GeneID	818408.
KEGG	ath:AT2G38280.

Organism-specific databases

TAIR	AT2G38280.
------	------------

Phylogenomic databases

eggNOG	COG1816.
HOGENOM	HOG000092200.
InParanoid	O80452.
KO	K01490.
OMA	HRVYSDN.
PhylomeDB	O80452.
ProtClustDB	PLN02768.

Enzyme and pathway databases

UniPathway	UPA00591 ; UER00663 .
SABIO-RK	O80452.

Miscellaneous databases

EvolutionaryTrace	O80452.
-------------------	---------

Gene expression databases

Genevestigator	O80452.
----------------	---------

Ontologies

PRO	O80452.
-----	---------

Family and domain databases

InterPro	IPR006650. A/AMP_deam_AS. IPR001365. A/AMP_deaminase_dom. IPR006329. AMP_deaminase. [Graphical view]
PANTHER	PTHR11359. PTHR11359. 1 hit.
Pfam	PF00962. A_deaminase. 1 hit. [Graphical view]
TIGRFAMs	TIGR01429. AMP_deaminase. 1 hit.
PROSITE	PS00485. A_DEAMINASE. 1 hit. [Graphical view]
ProteinNet	Search...

Fig. 21 Cross-references section of a UniProtKB entry. View of the “cross-references” section of the UniProtKB protein O80452

Table 1
Plant-specific cross-references present in UniProtKB

Database name and URL and goals	DR line format
GeneFarm [13] http://genoplante-info.infobiogen.fr/Genefarm/ Structural and functional annotation of <i>Arabidopsis thaliana</i> gene and protein families (see http://www.uniprot.org/database/DB-0032).	DR GeneFarm ; GeneID; FamilyID. <i>In UniProtKB/Swiss-Prot only</i>
Gramene; a comparative mapping resource for grains [14] http://www.gramene.org/ Curated, open-source, Web-accessible data resource for comparative genome analysis in the grasses (see http://www.uniprot.org/database/DB-0039).	DR Gramene ; UniProtKB_AC; -. <i>In UniProtKB/Swiss-Prot and UniProtKB/TrEMBL</i>
MaizeGenetics/GenomicsDatabase(MaizeGDB) [15] http://www.maizegdb.org/ Central repository for public maize information (see http://www.uniprot.org/database/DB-0058).	DR MaizeDB ; ProteinID; -. <i>In UniProtKB/Swiss-Prot only</i>
The Arabidopsis Information Resource (TAIR) [16] http://www.arabidopsis.org/index.jsp Searchable relational database on <i>Arabidopsis thaliana</i> , which includes many different molecular data types and provides a comprehensive resource for the scientific community (see http://www.uniprot.org/database/DB-0102).	DR TAIR ; Order_locus_name; -. <i>In UniProtKB/Swiss-Prot and UniProtKB/TrEMBL</i>

is indicated in the first column (see Fig. 21 i). A link to the relevant data in the cross-linked database is present in next columns. Plant specific databases that are currently cross-linked in UniProtKB entries are listed in Table 1. They have been chosen because of their content, their stability and their frequent updates. All of them give additional information about the protein and are linked back to UniProtKB.

The different subsections of the cross-references section are:


- (a) **2D gel databases**
- (b) **3D structure databases**; Cross-references to the PDB database (<http://www.rcsb.org/pdb/>) are present when protein structures are available. PDB cross-links contain information about the crystallographic method, the number of chains, and the range of residues present in the structure.
- (c) **Enzyme and pathway databases**
- (d) **Family and domain databases**
- (e) **Gene expression databases**

- (f) **Genome annotation databases**
 - (g) **Ontologies**
 - (h) **Organism-specific databases**
 - (i) **Phylogenomic databases**
 - (j) **Polymorphism databases**
 - (k) **Proteomic databases**
 - (l) **Protein-protein interaction databases**
 - (m) **Protein family/group databases**
 - (n) **PTM databases**
 - (o) **Sequence databases**; Cross-references to the EMBL database (<http://www.embl-heidelberg.de/>) are displayed in the same order as the corresponding references associated with a sequence submission. EMBL cross-links contain a nucleic acid sequence ID, a protein sequence ID and a molecule type to indicate the origin of the sequence (e.g., mRNA or Genomic_DNA) (*see Note 26*). When no coding sequence to translate the nucleic acid sequence into the protein sequence was provided by the submitters to the EMBL, the flag “No translation available” is present to replace the lacking protein sequence ID. When the sequence displayed in UniProt differs from the original EMBL sequence, a flag “Sequence problems” is added and the differences between the two sequences are summarized in the “Sequence” section.
 - (p) **Other**
12. “Publications”: (*see* Fig. 22 and http://www.uniprot.org/help/publications_section). This block lists all references used for the annotation of the protein entry. The first references are usually associated with sequence submission, followed by references providing other information concerning the function and structure of the protein. Each reference is numbered and contains title, authors, and conventional citation information for the reference, including cross-links to PubMed and digital object identifier (DOI), thus allowing retrieval of the electronic version of the article. In addition, an indication of what information was extracted from the article, strain and tissues used is also mentioned when available. In the case of references associated with a sequence submission, the sequenced molecule type is mentioned and, if relevant, the corresponding isoform is indicated. Each author name is linked to a UniProtKB query that retrieves all entries where that author is cited.
 13. “Entry information”: (*see* Fig. 23a and http://www.uniprot.org/help/entry_information_section). In addition to the

Publications

[« Hide 'large scale' publications](#)[Download](#)

1. **"EMBRYONIC FACTOR 1 encodes an AMP deaminase and is essential for the zygote to embryo transition in Arabidopsis."**
Xu J., Zhang H.-Y., Xie C.-H., Xue H.-W., Dijkhuis P., Liu C.-M.
[Plant J. 42:743-756\(2005\)](#) [[PubMed](#)] [[Europe PMC](#)] [[Abstract](#)]
Cited for: NUCLEOTIDE SEQUENCE [GENOMIC DNA], MUTAGENESIS OF ASP-598, FUNCTION, TISSUE SPECIFICITY, DEVELOPMENTAL STAGE.
Strain: *cv. Landsberg erecta*.

2. **"Sequence and analysis of chromosome 2 of the plant Arabidopsis thaliana."**
Lin X., Kaul S., Rounsley S.D., Shea T.P., Benito M.-I., Town C.D., Fujii C.Y., Mason T.M., Bowman C.L., Barnstead M.E., Feldblyum T.V., Buell C.R., Ketchum K.A., Lee J.J., Ronning C.M., Koo H.L., Moffat K.S., Cronin L.A.  Venter J.C.
[Nature 402:761-768\(1999\)](#) [[PubMed](#)] [[Europe PMC](#)] [[Abstract](#)]
Cited for: NUCLEOTIDE SEQUENCE [LARGE SCALE GENOMIC DNA].
Strain: *cv. Columbia*.

3. The Arabidopsis Information Resource (TAIR)
Submitted (APR-2011) to the EMBL/GenBank/DBJ databases
Cited for: GENOME REANNOTATION.
Strain: *cv. Columbia*.
...

8. **"Toward an interaction map of the two-component signaling pathway of Arabidopsis thaliana."**
Dortay H., Gruhn N., Pfeifer A., Schwerdtner M., Schmuelling T., Heyl A.
[J. Proteome Res. 7:3649-3660\(2008\)](#) [[PubMed](#)] [[Europe PMC](#)] [[Abstract](#)]
Cited for: INTERACTION WITH AHK4.
...

10. **"Crystallization and preliminary X-ray crystallographic analysis of adenosine 5'-monophosphate deaminase (AMPD) from Arabidopsis thaliana in complex with coformycin 5'-phosphate."**
Han B.W., Bingman C.A., Mahnke D.K., Sabina R.L., Phillips G.N. Jr.
[Acta Crystallogr. F 61:740-742\(2005\)](#) [[PubMed](#)] [[Europe PMC](#)] [[Abstract](#)]
Cited for: CRYSTALLIZATION, X-RAY CRYSTALLOGRAPHY (3.3 ANGSTROMS) OF 140-839 IN COMPLEX WITH COFORMYCIN 5'-PHOSPHATE AND ZINC IONS.

11. **"Membrane association, mechanism of action, and structure of Arabidopsis embryonic factor 1 (FAC1)."**
Han B.W., Bingman C.A., Mahnke D.K., Bannen R.M., Bednarek S.Y., Sabina R.L., Phillips G.N. Jr.
[J. Biol. Chem. 281:14939-14947\(2006\)](#) [[PubMed](#)] [[Europe PMC](#)] [[Abstract](#)]
Cited for: X-RAY CRYSTALLOGRAPHY (3.3 ANGSTROMS) OF 140-839 IN COMPLEX WITH COFORMYCIN 5'-PHOSPHATE; PHOSPHATE AND ZINC IONS, CATALYTIC ACTIVITY, SUBUNIT, COFACTOR, SUBCELLULAR LOCATION, ENZYME REGULATION, BIOPHYSICOCHEMICAL PROPERTIES.

Fig. 22 Publications section of a UniProtKB entry. View of the "publications" section of the UniProtKB protein 080452

primary accession number, a protein entry may contain one or more secondary accession numbers, which follow the primary accession number. These are usually accession numbers of UniProtKB/TrEMBL entries that have been merged into a single UniProtKB/Swiss-Prot entry. The history of the current protein entry give the date when the entry was first created, the date of last modification of the sequence and the date of last modification of annotation, respectively. The corresponding releases are also indicated. A quick access to this history is also available beneath the entry remote control (*see* Fig. 1 ix).

Entry information^a

a

Entry name ⁱ	AMPD_ARATH	
Accession ⁱ	Primary (citable) accession number: O80452 Secondary accession number(s): B9DFX9, Q56XX1, Q93ZR9	
Entry history ⁱ	Integrated into UniProtKB/Swiss-Prot:	May 30, 2006
	Last sequence update:	June 1, 2002
	Last modified:	October 16, 2013
	This is version 94 of the entry and version 2 of the sequence. [Complete history]	
Entry status ⁱ	Reviewed (UniProtKB/Swiss-Prot)	
Annotation program	Plant Protein Annotation Program	

Miscellaneous^b

b

Keywords - Technical term^t3D-structure, Allosteric enzyme, [Complete proteome](#), [Reference proteome](#)

Documents

[Arabidopsis thaliana](#)

Arabidopsis thaliana: entries and gene names

PATHWAY comments

Index of metabolic and biosynthesis pathways

PDB cross-references


Index of Protein Data Bank (PDB) cross-references

SIMILARITY comments

Index of protein domains and families

Similar proteins^c

c

90% Identity		50% Identity		Length	Cluster ID	Cluster name	Size	
Entry	Cluster member(s)	Organisms						
O80452	M4CLN9 R0FZK3 D7LLH0	Arabidopsis thaliana (Mouse-ear cress) Brassica rapa subsp. pekinensis (Chinese cabbage) (Brassica pekinensis) Capsella rubella Arabidopsis lyrata subsp. lyrata (Lyre-leaved rock-cress)		839	UniRef90_O80452	Cluster: AMP deaminase	4	

[Full view](#)

Fig. 23 Entry information, miscellaneous and similar proteins sections of a UniProtKB entry. View of the “information, miscellaneous and similar proteins” sections of the UniProtKB protein O80452

- “Miscellaneous”: (*see* Fig. 23b and http://www.uniprot.org/help/miscellaneous_section). Links to relevant documents (*see* **Note 2**) and keywords of the “Technical term” section are listed.
- “Similar proteins”: (*see* Fig. 23c and http://www.uniprot.org/help/similar_proteins_section). This section provides links to UniRef100, UniRef90, and UniRef50, corresponding to protein sequences sharing 100 %, 90 %, or 50 % identity, respectively. UniRef are sequence clusters, used to speed up sequence similarity searches (*see* **Note 4**).

4 Notes

1. The SIB (Switzerland, Geneva), in collaboration with the EBI (UK, Hinxton) and PIR (USA, Georgetown University Medical Center and National Biomedical Research Foundation), develop the UniProt protein resource that contain a Protein knowledgebase (UniProtKB), Sequence clusters (UniRef), and a sequence archive (UniParc).
2. For more information, *see* <http://www.uniprot.org/docs> and <http://www.expasy.org/sprot/userman.html>. UniProt propose also demonstration videos on its YouTube channel: <https://www.youtube.com/channel/UCkCR5RJZCZZoVTQzTYY92aw>.
3. For more information, *see* http://www.uniprot.org/manual/non_experimental_qualifiers.
4. The UniRef reference clusters combine closely related sequences into a single record in order to speed sequence similarity searches. The UniRef100 database combines identical sequences and subfragments of the UniProt Knowledgebase (from any species) and selected UniParc records into a single UniRef entry (<http://www.uniprot.org/help/uniref>). UniRef90 and UniRef50 yield a database size reduction of approximately 40 % and 65 %, respectively, providing for significantly faster sequence searches.
5. UniProtKB proteomes are listed at <http://www.uniprot.org/taxonomy/complete-proteomes>. Each protein of a reference organism has the keyword “Reference proteome” (*see* <http://www.uniprot.org/keywords/KW-1185>).
6. UniProt is currently hosted by a unified UniProt website <http://www.uniprot.org/>.
7. Major releases usually introduce important format changes. They are distinguishable from other releases by a new primary number followed by “.0”.
8. To download a local version of UniProtKB, use the web page <ftp://ftp.uniprot.org/pub>.
9. When a gene encodes different isoforms and/or when different protein sequences for the same gene of a given species (given cultivar/strain/isolate) are available, they are merged into a single UniProtKB entry (e.g., Jasmonic acid-amido synthetase JAR1, entry **Q9SKE2**).
10. Other tools and databases developed by the EBI and PIR are available at <http://www.ebi.ac.uk/services/> [17] and <http://pir.georgetown.edu/>, respectively.
11. For users of the Mozilla Web browser (<http://www.mozilla.org/>), the biobar navigation bar, dedicated to search into various

biological databases, is available at <https://addons.mozilla.org/en-US/firefox/addon/biobar/>. An ExPASy navigation bar is available at <http://expasybar.mozdev.org>, it allows searches to be performed in several databases hosted by ExPASy.

12. A complete documentation about BLAST parameters is available on the UniProt website at this address: <http://www.uniprot.org/help/sequence-searches>.
13. Your feedback is highly important and allows us to continuously improve our knowledgebase according to your needs.
14. UniProtKB accessions (AC) contain six characters and respect one of these regular expressions $[A-N,R-Z][0-9][A-Z][A-Z,0-9][A-Z,0-9][0-9]$ or $[O,P,Q][0-9][A-Z,0-9][A-Z,0-9][A-Z,0-9][0-9]$ (e.g., O80452). To face the fast increasing amount of new protein entries, an additional accession format extended to 10 alphanumeric characters for entries integrated after all 6 characters accessions will be used, possibly in 2014. The format of this new format will be $[A-N,R-Z][0-9][A-Z][A-Z,0-9][A-Z,0-9][0-9][A-Z][A-Z,0-9][A-Z,0-9][0-9]$. Both 6 and 10 characters accessions will coexist. All accessions are stable in time and should be used for UniProtKB protein citation.
15. It can also (but rarely) happen that the primary accession number becomes a secondary accession number (e.g., when an entry is split in two entries).
16. An accession number uniquely identifies an entry. If an entry is deleted, its AC will never be attributed to another entry.
17. A typical example is the annotation of N-glycosylation sites in the entries of non-cytoplasmic domains or proteins.
18. A typical example is the annotation of nuclear subcellular location in the entries of active transcription factors in eukaryotic organisms.
19. Exhaustive information about all cross-references present into UniProtKB (more than 140 in 2014) is available at <http://www.uniprot.org/database/> and <http://www.uniprot.org/docs/dbxref>.
20. Amino-acid residue numbering begins at the N-terminus of the precursor protein (the displayed sequence).
21. The description of the feature may contain a non-experimental qualifier (*see* http://www.uniprot.org/manual/non_experimental_qualifiers).
22. In the case of *Arabidopsis thaliana* and *Oryza sativa* (and in other organisms following the same standards), we use the following nomenclature according to the standard defined for *A. thaliana*: [first letter of the genus name]-[first letter of the

species name]-[chromosome number]-[g, for gene]-[locus number] (e.g., At1g15690, Os03g16440).

23. Currently, *Oryza sativa* has three different taxonomy identifiers in UniProtKB/TrEMBL: **39947** for japonica cultivars, **39946** for indica cultivars, and **4530** for unspecified rice cultivars. In UniProtKB/Swiss-Prot, when possible, cultivars are specified for each reference related to a sequence deposition.
24. The family classification is exclusively based on sequence similarities, not on functions.
25. The algorithm to compute the CRC64 is described in the ISO 3309 standard [12].
26. Additional qualifiers may be present: ALT_SEQ, ALT_INIT, ALT_TERM, or ALT_FRAME. These are used in the case of discrepancies between the EMBL derived CDS and the displayed protein sequence. These may be due to gross differences in the predicted CDS sequence (arising from the failure to correctly predict all exons for a given gene for instance), incorrect selection of the initiating methionine, and termination of the sequence or frameshifts, respectively. For more details, see the documentation (http://www.uniprot.org/help/sequence_caution).

Acknowledgments

UniProt is mainly supported by the National Institutes of Health (NIH) grant 1 U41 HG006104. Additional support for the EBI's involvement in UniProt comes from the NIH grant 2P41 HG02273. Swiss-Prot activities at the SIB are supported by the Swiss Federal Government through The State Secretariat for Education, Research and Innovation SERI. PIR's UniProt activities are also supported by the NIH grants 5R01GM080646-07, 3R01GM080646-07S1, 5G08LM010720-03, and 8P20GM103446-12, and the National Science Foundation (NSF) grant DBI-1062520. We would like to thank all Swiss-Prot curators and developers for their contribution to the expert annotation of proteins and their critical reading of the manuscript.

References

1. The UniProt Consortium (2014) Activities at the Universal Protein Resource (UniProt). *Nucleic Acids Res* 42(Database issue):D191–D198
2. Bairoch A, Boeckmann B, Ferro S, Gasteiger E (2004) Swiss-Prot: juggling between evolution and stability. *Brief Bioinform* 5:39–55
3. Boeckmann B, Bairoch A, Apweiler R, Blatter M-C, Estreicher A, Gasteiger E, Martin MJ, Michoud K, O'Donovan C, Phan I, Pilbout S, Schneider M (2003) The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Res* 31: 365–370
4. Hunter S, Jones P, Mitchell A, Apweiler R, Attwood TK, Bateman A, Bernard T, Binns D, Bork P, Burge S, de Castro E, Coghill P, Corbett M, Das U, Daugherty L, Duquenne L, Finn RD, Fraser M, Gough J, Haft D, Hulo N,

- Kahn D, Kelly E, Letunic I, Lonsdale D, Lopez R, Madera M, Maslen J, McAnulla C, McDowall J, McMenamin C, Mi H, Mutow-Muellenet P, Mulder N, Natale D, Orengo C, Pesseat S, Punta M, Quinn AF, Rivoire C, Sangrador-Vegas A, Selengut JD, Sigrist CJ, Scheremetjew M, Tate J, Thimmajananthan M, Thomas PD, Wu CH, Yeats C, Yong SY (2012) InterPro in 2011: new developments in the family and domain prediction database. *Nucleic Acids Res* 40(Database issue): D306–D312
5. Schneider M, Lane L, Boutet E, Lieberherr D, Tognolli M, Bougueleret L, Bairoch A (2009) The UniProtKB/Swiss-Prot knowledgebase and its Plant Proteome Annotation Program. *J Proteomics* 72(3):567–573
 6. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ (1990) Basic local alignment search tool. *J Mol Biol* 215:403–410
 7. Gattiker A, Gasteiger E, Bairoch A (2002) ScanProsite: a reference implementation of a PROSITE scanning tool. *Appl Bioinforma* 1:107–108
 8. Gasteiger E, Gattiker A, Hoogland C, Ivanyi I, Appel RD, Bairoch A (2003) ExpASY: the proteomics server for in-depth protein knowledge and analysis. *Nucleic Acids Res* 31:3784–3788
 9. Gasteiger E, Hoogland C, Gattiker A, Duvaud S, Wilkins MR, Appel RD, Bairoch A (2005) Protein identification and analysis tools on the ExpASY Server. In: Walker JM (ed) *The proteomics protocols handbook*. Humana, Totowa, NJ, pp 571–607
 10. Dimmer EC, Huntley RP, Alam-Faruque Y, Sawford T, O'Donovan C, Martin MJ et al (2012) The UniProt-GO Annotation database in 2011. *Nucleic Acids Res* 40:D565–D570
 11. Bairoch A (2000) The ENZYME database in 2000. *Nucleic Acids Res* 28:304–305
 12. Press WH, Flannery BP, Teukolsky SA, Vetterling WT (1993) Numerical recipes in C, 2nd edn. Cambridge University Press, Cambridge, pp 896–902
 13. Aubourg S, Brunaud V, Bruyere C, Cock M, Cooke R, Cottet A, Couloux A, Dehais P, Deleage G, Duclert A, Echeverria M, Eschbach A, Falconet D, Filippi G, Gaspin C, Geourjon C, Grienenberger J-M, Houline G, Jamet E, Lechauve F, Leleu O, Leroy P, Mache R, Meyer C, Nedjari H, Negrutiu I, Orsini V, Peyretailade E, Pommier C, Raes J, Risler J-L, Riviere S, Rombauts S, Rouze P, Schneider M, Schwob P, Small I, Soumayet-Kampetenga G, Stankovski D, Toffano C, Tognolli M, Caboche M, Lechardy A (2005) GeneFarm, structural and functional annotation of Arabidopsis gene and protein families by a network of experts. *Nucleic Acids Res* 33:D641–D646
 14. Ware DH, Jaiswal P, Ni J, Yap IV, Pan X, Clark KY, Teytelman L, Schmidt SC, Zhao W, Chang K, Cartinhour S, Stein LD, McCouch SR (2002) Gramene, a tool for grass genomics. *Plant Physiol* 130:1606–1613
 15. Lawrence CJ, Dong Q, Polacco ML, Seigfried TE, Brendel V (2004) MaizeGDB, the community database for maize genetics and genomics. *Nucleic Acids Res* 32(Database issue): D393–D397
 16. Rhee SY, Beavis W, Berardini TZ, Chen G, Dixon D, Doyle A, Garcia-Hernandez M, Huala E, Lander G, Montoya M, Miller N, Mueller LA, Mundodi S, Reiser L, Tacklind J, Weems DC, Wu Y, Xu I, Yoo D, Yoon J, Zhang P (2003) The Arabidopsis Information Resource (TAIR): a model organism database providing a centralized, curated gateway to Arabidopsis biology, research materials and community. *Nucleic Acids Res* 31:224–228
 17. Harte N, Silventoinen V, Quevillon E, Robinson S, Kallio K, Fustero X, Patel P, Jokinen P, Lopez R (2004) European Bioinformatics Institute. Public web-based services from the European Bioinformatics Institute. *Nucleic Acids Res* 32(Web Server issue):W3–W9

KEGG Bioinformatics Resource for Plant Genomics and Metabolomics

Minoru Kanehisa

Abstract

In the era of high-throughput biology it is necessary to develop not only elaborate computational methods but also well-curated databases that can be used as reference for data interpretation. KEGG (<http://www.kegg.jp/>) is such a reference knowledge base with two specific aims. One is to compile knowledge on high-level functions of the cell and the organism in terms of the molecular interaction and reaction networks, which is implemented in KEGG pathway maps, BRITE functional hierarchies, and KEGG modules. The other is to expand knowledge on genes and proteins involved in the molecular networks from experimentally observed organisms to other organisms using the concept of orthologs, which is implemented in the KEGG Orthology (KO) system. Thus, KEGG is a generic resource applicable to all organisms and enables interpretation of high-level functions from genomic and molecular data. Here we first present a brief overview of the entire KEGG resource, and then give an introduction of how to use KEGG in plant genomics and metabolomics research.

Key words KEGG pathway map, KEGG Mapper, Plant genomeannotation, Plant metabolism, Phytochemical compounds

1 Introduction

KEGG (Kyoto Encyclopedia of Genes and Genomes) is a database resource for understanding high-level functions and utilities of the biological system, such as the cell, the organism, and the ecosystem, from genomic and molecular level information [1, 2]. It is widely used for biological interpretation of large-scale molecular datasets generated by genomics, metabolomics, and other high-throughput experimental technologies. The basic idea of KEGG is to develop a knowledge base that would enable conversion of molecular building blocks of genes and chemicals to molecular networks of biological pathways, which are considered as wiring diagrams of biological systems for performing specific functions. Thus, KEGG is a dictionary (or an encyclopedia) for translation of genes in the genome or metabolites in the metabolome into specific

pathways, enabling interpretation of cellular functions and organismal behaviors.

Subheading 2 of this paper describes an overview of KEGG, which is a general resource applicable to all organisms from prokaryotes to eukaryotes. Subheading 3 describes how the KEGG resource and associated bioinformatics methods can be used in plant genomics and metabolomics research. Plants are known to produce diverse chemical substances including those with medicinal and nutritional values. Our knowledge on biosynthetic pathways of plant natural products is largely incomplete, but the genomic and metabolomic information is expected to give clues to missing enzymes and reactions for biosynthesis. An increasing number of plant genome sequences may also uncover more basic architectural principles of biosynthetic pathways for generating chemical diversity of natural products. Subheading 4 discusses the use of newly developed KEGG modules and reaction modules [1] for predictive analysis of plant metabolic pathways.

2 Materials

2.1 *KEGG Molecular Networks*

KEGG is an integrated resource consisting of 16 main databases, which are categorized into systems information, genomic information, chemical information and health information (Table 1). Among them the most unique ones are the three databases for molecular networks in the systems information category. The KEGG PATHWAY database consists of manually drawn KEGG pathway maps representing our knowledge on the molecular interaction and reaction networks for metabolism, various other cellular processes, organismal systems, and human diseases. The KEGG BRITE database is a collection of hierarchical classification systems (ontologies) for various biological objects including genes, proteins, small molecules, drugs, diseases, and organisms. While KEGG pathway maps are limited to molecular interactions and reactions, BRITE hierarchies incorporate many different types of relationships. The KEGG MODULE database is a collection of manually defined functional units of genes, which are used primarily for annotation and biological interpretation of sequenced genomes.

2.2 *KEGG Orthology (KO)-Based Genome Annotation*

The three databases for molecular networks are developed by capturing experimental knowledge on systemic functions from published literature and by organizing this knowledge in computable forms, namely, as KEGG pathway maps, BRITE functional hierarchies and KEGG modules. At the same time this knowledge is expanded from experimentally observed organisms to other organisms through the KEGG Orthology (KO) system. The KEGG ORTHOLOGY database in the genomic information category (Table 1) consists of manually defined ortholog groups or

Table 1
The databases of KEGG

Category	Database name	Content
Systems information	KEGG PATHWAY	KEGG pathway maps
	KEGG BRITE	BRITE functional hierarchies
	KEGG MODULE	KEGG modules
Genomic information	KEGG ORTHOLOGY	KEGG Orthology (KO) groups
	KEGG GENOME	KEGG organisms with complete genomes
	KEGG GENES	Gene catalogs of complete genomes
Chemical information	KEGG COMPOUND	Metabolites and other small molecules
	KEGG GLYCAN	Glycans
	KEGG REACTION	Biochemical reactions
	KEGG RPAIR	Reactant pairs
	KEGG RCLASS	Reaction class
	KEGG ENZYME	Enzyme nomenclature
Health information	KEGG DISEASE	Human diseases
	KEGG DRUG	Approved drugs
	KEGG DGROUP	Drug groups
	KEGG ENVIRON	Crude drugs and health-related substances

KO groups. The definition of KO groups is an integral part of developing KEGG PATHWAY, BRITE, and MODULE databases, as well as genome annotation in KEGG. Gene/protein nodes of KEGG molecular networks are defined by KO identifiers called K numbers, and appropriate groupings of orthologs are continuously evaluated for all available genomes. The KEGG GENOME database contains organism level information for completely sequenced genomes, and the KEGG GENES database is a collection of gene catalogs for these genomes, which are given KO based annotations (K number assignments).

The fact that functional information is associated with ortholog groups is another unique aspect of the KEGG resource. The sequence similarity based inference as a generalization of limited amount of experimental evidence is predefined in KEGG. In other databases such as UniProt [3] functional information is associated with individual proteins or genes. The sequence similarity search against such databases will require processing of search results, which may contain a large amount of data as the database size increases. In contrast, the sequence similarity search against KEGG

GENES is a search for most appropriate K numbers, which can easily be computerized as implemented in the KOALA [1] and KAAS [4] programs. Once the K numbers are assigned to genes in the genome, the KEGG molecular networks are automatically reconstructed, enabling biological interpretation of high-level functions.

2.3 KEGG LIGAND

The databases in the chemical information category (Table 1) are collectively called KEGG LIGAND. Traditionally, metabolism has been a major content of the KEGG resource. Metabolism represents a chemical system for generating all necessary chemical substances through chemical reactions, as well as a genetic system of genome-encoded enzymes that catalyze chemical reactions. KEGG LIGAND, especially its reaction dataset, has been developed to capture empirical knowledge on this dual aspect of metabolism. The KEGG REACTION database contains all known enzymatic reactions taken from the Enzyme Nomenclature in the KEGG ENZYME database and the metabolic pathway section of the KEGG PATHWAY database. Reaction data are processed using the concept of reactant pairs in the KEGG RPAIR database to generate reaction class entries in the KEGG RCLASS database [5]. Reaction class is like an ortholog group of reactions, which exhibits the same local structure transformation patterns but may involve different metabolites with different overall structures. Empirical relationships are being established between reaction orthologs in KEGG RCLASS (RC) and gene orthologs in KEGG ORTHOLOGY (KO), specifically in terms of reaction modules and KEGG modules [6] (*see* Subheading 4).

The other databases in the chemical information category are KEGG COMPOUND and KEGG GLYCAN, which are chemical structure databases for small molecules and glycans, respectively. KEGG COMPOUND originally contained substrates and products of enzymatic reactions, but it has been extended to include chemical compounds with any biological roles, including xenobiotic compounds.

2.4 KEGG MEDICUS

The databases in the health information category (Table 1) are KEGG DISEASE, DRUG, DGROUP, and ENVIRON, which together with outside databases of drug labels (package inserts) constitute the KEGG MEDICUS resource for translational bioinformatics [1]. Here, diseases are viewed as perturbed states of the biological system caused by genetic perturbations such as germline and somatic mutations, and environmental perturbations such as carcinogens and pathogens. In addition, drugs are different types of perturbations that would correct perturbed states of the biological system. The KEGG DISEASE database is a collection of human diseases represented by such perturbations. The KEGG DRUG

database is a comprehensive collection of approved drugs in Japan, the USA, and Europe unified based on the chemical structures and/or the chemical components, and associated with target, metabolizing enzyme, and other molecular interaction network (perturbation) information.

In order to better characterize how drugs affect the molecular network, or how drugs interact with other molecules in the molecular network, the KEGG DGROUP (DG) database for drug groups is being developed. In a similar way as individual instances of genes and reactions are generalized by KO groups and RC groups, respectively, individual instances of drugs are now generalized by DG groups. The KEGG ENVIRON database is a collection of crude drugs, essential oils, and other health-promoting substances, which are mostly natural products of plants. This collection supplements the collection of KEGG DRUG containing only the approved drugs.

2.5 KEGG Object Identifiers

KEGG objects are biological entities from molecular to higher levels that are represented as database entries in KEGG. Among the 16 databases shown in Table 1, KEGG GENES is derived from public sequence databases, mostly RefSeq and GenBank, and given gene annotations (K number assignments). KEGG ENZYME is derived from ExplorEnz database [7] and given annotations of reactions and enzyme genes. The other 14 are all KEGG original databases that have been developed with extensive manual curation. The KEGG object identifiers (database entry names) for the 14 databases are five-digit numbers preceded by database dependent prefixes as shown in Table 2. The K number identifier for KO entries has already been mentioned. Other identifiers include the C number for small molecules, the D number for drugs and the H number for diseases. Each T number identifier for a genome entry has an alias of three- or four-letter organism code, such as T00041 and ath for *Arabidopsis thaliana*. The KEGG GENES identifier takes the form of “org:gene” where “org” is the organism code and “gene” is the gene identifier in the original database, such as locus_tag or Gene ID in RefSeq. Furthermore, the identifiers for molecular networks have variants of organism-specific versions by replacing the prefix. For example, map00010 is the manually drawn pathway map for glycolysis, and hsa00010 is the human pathway map that is computationally generated. Similarly, ko02000 is the manually developed Brite hierarchy for transporters, and ath02000 is the computationally generated version for *Arabidopsis thaliana*. The KEGG object identifiers are uniquely defined in the entire KEGG resource and searchable in Google and other web search engines.

Table 2
KEGG object identifiers

Database	Prefix or db:entry	Example	URL
KEGG PATHWAY	map, ko, ec, rn, (org)	map00940	www.kegg.jp/pathway/map00940
KEGG BRITE	br, jp, ko, (org)	ko00199	www.kegg.jp/brite/ko00199
KEGG MODULE	M, (org)_M	M00039	www.kegg.jp/module/M00039
KEGG ORTHOLOGY	K	K00487	www.kegg.jp/entry/K00487
KEGG GENOME	T	T00041 (ath)	www.kegg.jp/entry/T00041
KEGG GENES	(org:gene)	ath:AT2G30490	www.kegg.jp/entry/ath:AT2G30490
KEGG COMPOUND	C	C02646	www.kegg.jp/entry/C02646
KEGG GLYCAN	G	G12794	www.kegg.jp/entry/G12794
KEGG REACTION	R	R02253	www.kegg.jp/entry/R02253
KEGG RPAIR	RP	RP02071	www.kegg.jp/entry/RP02071
KEGG RCLASS	RC	RC00490	www.kegg.jp/entry/RC00490
KEGG ENZYME	(ec:number)	ec:1.14.13.11	www.kegg.jp/entry/ec:1.14.13.11
KEGG DISEASE	H	H00020	www.kegg.jp/entry/H00020
KEGG DRUG	D	D08086	www.kegg.jp/entry/D08086
KEGG DGROUPE	DG	DG00726	www.kegg.jp/entry/DG00726
KEGG ENVIRON	E	E00259	www.kegg.jp/entry/E00259

Each T number is given three- or four-letter organism code, which is used in GENES identifier

3 Methods

3.1 Plant Datasets in KEGG

The KEGG resource is accessible either at the KEGG website (www.kegg.jp) or the GenomeNet mirror site (www.genome.jp/kegg/). KEGG is a general resource for all organisms, and it may be difficult to find plant-specific datasets without knowing the architecture of the KEGG website. Table 3 shows some examples of plant datasets, which include the pathway maps of plant secondary metabolism, the genomes of plants and related organisms, such as plant pathogens and plant symbionts, and the phytochemical compounds especially with health-related values.

The architecture of the KEGG website is illustrated in Fig. 1. The first layer is the top page (url shown above) linking to main databases, selected computational tools, and documents on the left

panel. The second layer is color coded in yellow containing the KEGG Table of Contents (KEGG2) page for a more detailed version of links to all databases and computational tools, the KEGG PATHWAY page for the list of KEGG pathway maps and the KEGG BRITE page for the list of BRITE functional hierarchies. The examples shown in Table 3 can be accessed from the PATHWAY and BRITE pages. The second layer menu bar also contains links to the third layer menus, which are distinguished by different color coding. The MODULE link in the yellow menu bar leads to a different menu bar in green (systems information), the KO, GENOME, and GENES links to a menu bar in brown or red (genomic information), the LIGAND link to a menu bar in blue (chemical information), and the DISEASE and DRUG links to a menu bar in purple (health information).

Table 3
Examples of KEGG pathway maps and BRITE hierarchies for plant research

Content	URL
Pathway maps	
Biosynthesis of secondary metabolites	www.kegg.jp/pathway/map01110
Metabolism of terpenoids and polyketides	www.kegg.jp/kegg/pathway.html#pk
Biosynthesis of other secondary metabolites	www.kegg.jp/kegg/pathway.html#secondary
Biosynthesis of plant secondary metabolites	www.kegg.jp/pathway/map01060
Brite hierarchies	
KEGG organisms + keywords	www.kegg.jp/brite/br08601_key
Plant classification	www.kegg.jp/brite/br08603
Photosynthetic organisms	www.kegg.jp/brite/br08604
Endosymbionts	www.kegg.jp/brite/br08607
Plant pathogens	www.kegg.jp/brite/br08605
Phytochemical compounds	www.kegg.jp/brite/br08003
Crude drugs	www.kegg.jp/brite/br08305
Essential oils	www.kegg.jp/brite/br08321
Medicinal herbs	www.kegg.jp/brite/br08322
Major components of natural products	www.kegg.jp/brite/br08323

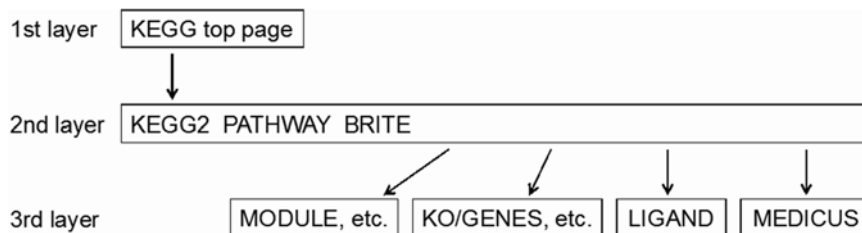


Fig. 1 The architecture of the KEGG website including the top page (first layer) at <http://www.kegg.jp/> and the Table of Contents page (second layer) at <http://www.kegg.jp/kegg/kegg2.html>

3.2 Plant Pathway Maps

The KEGG PATHWAY database is a collection of KEGG pathway maps. As an example, let us examine map00940 for phenylpropanoid biosynthesis (www.kegg.jp/pathway/map00940) shown in Fig. 2. Phenylpropanoids are a group of plant secondary metabolites derived from phenylalanine. This particular pathway is drawn as a metabolic grid, where the horizontal direction indicates successive hydroxylation and methylation steps of the aromatic ring (*see Note 1*) while the vertical direction indicates reduction of the CoA activated phenylpropane unit. The manually drawn pathway map, whose identifier is prefixed by “map,” is called a reference pathway map, which is a generic map for multiple organisms represented by KO (KEGG Orthology) groups. By using the genomic constraint of what genes are present in the genome, the organism-specific pathway map with a three- or four-letter organism code as a prefix is computationally generated. The KO nodes are converted to gene nodes and highlighted in green when corresponding genes are found in the genome. For example, use the popup menu or the “Organism menu” to obtain the organism-specific pathway ath00940 for *Arabidopsis thaliana* (www.kegg.jp/pathway/ath00940). In addition to the coloring, linked objects are different in differently prefixed maps. For example, the rectangular node marked with 1.14.13.11 is trans-cinnamate 4-monooxygenase, which is linked to a K number (KO) entry in the reference pathway (map00940) and a gene entry in the *Arabidopsis* pathway (ath00940) as shown in Fig. 2. There are also additional reference pathway maps prefixed with “ko,” “ec,” and “rn” for KO groups, enzymes and reactions, respectively, and highlighted in blue. They are computationally generated from the “map” reference pathway map.

The genomic constraint often clarifies specific routes in the pathway map, but it is not apparently the case for phenylpropanoid biosynthesis where the same KO groups are assigned to related reactions in the grid. In contrast, Fig. 2 contains highlighted nodes for specific reaction steps leading to three monolignols, *p-coumaryl* alcohol, coniferyl alcohol, and sinapyl alcohol, which are defined as part of the KEGG module M00039 for monolignol biosynthesis (www.kegg.jp/pathway/map00940+M00039) in the KEGG MODULE database (*see Note 2*). The three monolignols are

represented by circular nodes, and Fig. 4a shows the KEGG COMPOUND entry for one of them, *p*-coumaryl alcohol (www.kegg.jp/entry/C02646).

3.3 Plant-Related Brite Hierarchies

The KEGG BRITE database is a collection of hierarchical classification systems, not only for genes and proteins, but also chemical compounds and other biological entities. The reverse links to these classification systems are incorporated in the Brite field of individual entries as shown in Fig. 3 for KO and gene entries and in Fig. 4a for a compound entry. Here again only the reference Brite hierarchy with prefix “ko” is manually developed and the organism-specific versions are computationally generated. As indicated in the two Brite fields of Fig. 3, ko00199 for cytochrome P450 (www.kegg.jp/brite/ko00199) is used to generate ath00199 as its Arabidopsis version (www.kegg.jp/brite/ath00199). The reference Brite hierarchies other than those for genes and proteins are prefixed with “br,” such as br08003 for phytochemical compounds (www.kegg.jp/brite/br08003) shown in Fig. 4b and br08605 for plant pathogens (www.kegg.jp/brite/br08605). These non-gene/protein Brite hierarchies are developed with the aim of better understanding of genes and proteins. It is expected that accumulated knowledge of well-characterized phytochemical compounds would help uncover biosynthetic pathways and responsible enzyme genes encoded in the plant genome. Appropriate metadata annotation of sequenced genomes would help better understand organism-level interactions and responsible genes involving plants, pathogens, and endosymbionts.

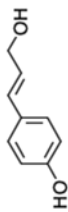
3.4 Using Genome Annotation Service

Genome annotation in KEGG is based on the sequence similarity search against the KEGG GENES database that is associated with the K number (KO) annotation. As more plant genomes are stored in KEGG GENES (there are 41 as of August 2014), this method will become more accurate for plants. Two web services are available for automatic annotation of users' genome or metagenome sequence data. One is newly released BlastKOALA (www.kegg.jp/blastkoala/) at the KEGG website and the other is KAAS (www.genome.jp/tools/kaas/) [4] at the GenomeNet mirror site (*see Note 3*). Both assign K numbers to genes in the user's genome, which then allow pathway mapping and other methods to infer high-level functions. Here a brief description is given for how to use BlastKOALA.

KOALA (KEGG Orthology And Links Annotation) is KEGG's internal annotation tool for K number assignment of KEGG GENES using SSEARCH computation results in KEGG SSDB [1]. BlastKOALA accepts users' amino acid sequence data and assigns K numbers by the same algorithm after BLAST search against KEGG GENES. The BlastKOALA top page is a query form for (1) FASTA-formatted query sequence data file, (2)

a **KEGG** **Phytochemical Compounds**

COMPOUND: C02646

Entry	C02646	Compound
Name	4-Coumaryl alcohol; 4-Hydroxycinnamyl alcohol; p-Coumaryl alcohol	
Formula	C9H10O2	
Exact mass	150.0681	
Mol weight	150.1745	
Structure		
Reaction	R04005 R04006 R04007 R06580 R07437 R10254	
Pathway	map00940 Phenylpropanoid biosynthesis map01061 Biosynthesis of phenylpropanoids map01100 Metabolic pathways map01110 Biosynthesis of secondary metabolites	
Module	M00039 Monolignol biosynthesis, phenylalanine/tyrosine => monolignol	
Enzyme	1.1.1.195 1.11.1.7 1.11.1.21 2.3.1.- 2.4.1.111 3.2.1.126	
Brite	Phytochemical compounds [BR:br08003] Phenylpropanoids Monolignols Paracoumaryl alcohol derivatives C02646 4-Coumaryl alcohol BRITE hierarchy	
Other DBs	CAS: 3690-05-9 PubChem: 5623 ChEBI: 28386 64555 KNAPSACK: C00000613 C00031420 3DMET: B00473 NIKKAJI: J117.130E	
LinkDB	All DBs	
KCF data	Show	

b

KEGG **Phytochemical Compounds**

[Brite menu | Download htctx]

Phytochemical compounds :

- ▼ ▼ ▼ One-click mode
- ▶ **Alkaloids**
- ▶ **Flavonoids**
- ▼ **Phenylpropanoids**
 - ▼ Monolignols
 - ▶ Caffeate derivatives
 - ▶ Coniferyl alcohol derivatives
 - ▼ Paracoumaryl alcohol derivatives
 - C02646 4-Coumaryl alcohol
 - C00811 4-Coumarate
 - C00223 p-Coumaroyl-CoA
 - C02947 4-Coumaroylshikimate
 - C05608 4-Hydroxycinnamyl aldehyde
 - C05855 4-Hydroxycinnamyl alcohol 4-D-glucoside
 - ▶ Sinapate derivatives
 - ▶ Others
 - ▶ Lignans
 - ▶ Coumarins
- ▶ **Skimate / acetate-malonate pathway derived compounds**
- ▶ **Terpenoids**
- ▶ **Polyketides**
- ▶ **Fatty acids related compounds**
- ▶ **Amino acid related compounds**
- ▶ **Others**

[BRITE | KEGG2 | KEGG]

Fig. 4 (a) The KEGG COMPOUND entry for 4-coumaryl alcohol (C02646). **(b)** The BRITE hierarchy for phytochemical compound (br08003), which is highlighted with C02646

a 

ORTHOLOGY: K00487

Entry Name	K00487 CYP73A
Definition	trans-cinnamate 4-monoxygenase [EC:1.14.13.11]
Pathway	ko03360 Phenylalanine metabolism ko00940 Phenylpropanoid biosynthesis ko00941 Flavonoid biosynthesis ko00945 Stilbenoid, diarylheptanoid and gingerol biosynthesis ko01220 Degradation of aromatic compounds M00039 Monoglignol biosynthesis, phenylalanine/tyrosine => M00137 Flavonone biosynthesis, phenylalanine => naringenin M00350 Capsaicin biosynthesis, L-Phenylalanine => Capsaicin
Module	
Brite	KEGG Orthology (KO) [BR:ko00001] Metabolism Overview 01220 Degradation of aromatic compounds K00487 CYP73A; trans-cinnamate 4-monoxygenase Amino acid metabolism K00487 CYP73A; trans-cinnamate 4-monoxygenase Biosynthesis of other secondary metabolites 00940 Phenylpropanoid biosynthesis K00487 CYP73A; trans-cinnamate 4-monoxygenase 00945 Stilbenoid, diarylheptanoid and gingerol biosynthesis K00487 CYP73A; trans-cinnamate 4-monoxygenase 00941 Flavonoid biosynthesis K00487 CYP73A; trans-cinnamate 4-monoxygenase KEGG modules [BR:ko00002] Pathway module Nucleotide and amino acid metabolism Aromatic amino acid metabolism M00350 Capsaicin biosynthesis, L-Phenylalanine => Capsaicin K00487 CYP73A; trans-cinnamate 4-monoxygenase Secondary metabolism Biosynthesis of secondary metabolites M00039 Monoglignol biosynthesis, phenylalanine / tyrosine => mon K00487 CYP73A; trans-cinnamate 4-monoxygenase M00137 Flavonone biosynthesis, phenylalanine => naringenin K00487 CYP73A; trans-cinnamate 4-monoxygenase Enzymes [BR:ko01000] 1. Oxidoreductases 1.14 Acting on paired donors with incorporation of molecular oxygen 1.14.13 With NADH or NADPH as one donor, and incorporation of one 1.14.13.11 trans-cinnamate 4-monoxygenase K00487 CYP73A; trans-cinnamate 4-monoxygenase Cytochrome P450 [BR:ko00199] CYP73 family K00487 CYP73A; trans-cinnamate 4-monoxygenase
Other Dbs	RN: R02253 R08815 GO: 0016710
Genes	ATH: AT2G30490(C4H) ALY: ABALPDB3BT_481961 CBB: CABUA_V1002304489 EUS: EUTSA_V1001654489 EUTSA_V1001654589 CIT: 10257934(C4H1)_102578013(C4H2) CIC: CICLE_V1000092189 CICLE_V1000812599 TCC: TCM_039725 TCM_042294 TCM_045554

b 

Arabidopsis thaliana (thale cress): AT2G30490

Entry Name	AT2G30490 CDS F00041
Gene name	C4H
Definition	trans-cinnamate 4-monoxygenase
Orthology	K00487 trans-cinnamate 4-monoxygenase [EC:1.14.13.11]
Organism	ath Arabidopsis thaliana (thale cress)
Pathway	ath00360 Phenylalanine metabolism ath00940 Phenylpropanoid biosynthesis ath00941 Flavonoid biosynthesis ath00945 Stilbenoid, diarylheptanoid and gingerol biosynthesis ath01100 Metabolic pathways ath01110 Biosynthesis of secondary metabolites ath01220 Degradation of aromatic compounds ath_M00039 Monoglignol biosynthesis, phenylalanine/tyrosine => ath_M00137 Flavonone biosynthesis, phenylalanine => naringenin ath_M00350 Capsaicin biosynthesis, L-Phenylalanine => Capsaicin
Module	
Brite	KEGG Orthology (KO) [BR:ath00001] Metabolism Overview 01220 Degradation of aromatic compounds AT2G30490 (C4H) Amino acid metabolism 00360 Phenylalanine metabolism AT2G30490 (C4H) Biosynthesis of other secondary metabolites 00940 Phenylpropanoid biosynthesis AT2G30490 (C4H) 00945 Stilbenoid, diarylheptanoid and gingerol biosynthesis AT2G30490 (C4H) 00941 Flavonoid biosynthesis AT2G30490 (C4H) Enzymes [BR:ath01000] 1. Oxidoreductases 1.14 Acting on paired donors with incorporation of molecular oxygen 1.14.13 With NADH or NADPH as one donor, and incorporation of one 1.14.13.11 trans-cinnamate 4-monoxygenase AT2G30490 (C4H) Cytochrome P450 [BR:ath00199] Cytochrome P450, plant type CYP73 family AT2G30490 (C4H)
SSDB	BRITE hierarchy
Motif	Pfam: P450
Other Dbs	NCBI-GI: 15274514 NCBI-Genes: 817559 MIPS: AT2G30490.1 TAIR: AT2G30490 UniProt: P92994 B1GV49
LinkDB	All Dbs
Position	2
AA seq	505 aa AA seq DB search MOLLESLAVFVAVLATYISKLRGKKLKFPGPIPIFGNWLQVGDILHRLVY YAKFDFDLFLNKKQRIIVVSSPDLTKVLLTQGVFSGSRTRVVFDFGKQDQVFT

Fig. 3 (a) The KEGG ORTHOLOGY (KO) entry for trans-cinnamate 4-monoxygenase (K00487). **(b)** The KEGG GENES entry for trans-cinnamate 4-monoxygenase in Arabidopsis thaliana (ath:AT2G30490)

taxonomy group used for adjusting sequence similarity scores, (3) KEGG database to be searched such as “all organisms” and “eukaryotes only,” and (4) your e-mail address. Once the form is submitted you will immediately receive an email. You must click on the link in the email to initiate your job. Once the job is completed you will receive another email containing the link to access the result page. You can now view or download the list of K numbers assigned and perform KEGG Mapper analyses using the Reconstruct Pathway, Reconstruct Brite and Reconstruct Module tools (*see* next section).

3.5 Using KEGG Mapper

KEGG mapping is the process to map molecular datasets, especially large-scale datasets generated by genomics, transcriptomics, metabolomics, and other high-throughput experiments, to the reference knowledge base of KEGG molecular networks (KEGG pathway maps, BRITE functional hierarchies and KEGG modules). It is not simply an enrichment process; rather it is a set operation to convert molecular datasets to molecular networks enabling biological interpretation of cellular and organism-level functions. The web interface for KEGG mapping is called KEGG Mapper (www.kegg.jp/kegg/mapper.html) consisting of various tools as summarized in Table 4. The mapping is based on the KEGG object identifiers (Table 2) such as K numbers for KO entries, org:gene identifiers for KEGG GENES entries and C numbers for KEGG COMPOUND entries. In other words, external identifiers of genes, proteins, small molecules, etc. must first be converted to the KEGG object identifiers.

There are four types of KEGG Mapper tools. First, the basic Search tools (Search Pathway, Search Brite and Search Module) are used to search query data against KEGG pathways, Brite hierarchies and KEGG modules. The found data are marked in red. Second, the Search&Color tools (Search&Color Pathway, Search&Color Brite and Search&Color Module) are variants of the basic Search tools, where query data may be associated with specification of any background and foreground colors. The found data are highlighted with this color specification. The tools in the third type (Color Pathway, Color Pathway WebGL, and Join Brite) do not involve database search and are used with a given KEGG pathway map or a given Brite hierarchy. The Color Pathway tools accept numerical values that may be converted to gradation of specified coloring or 3D projection of bar charts in WebGL graphics. The Color Pathway tools also accept tab-delimited multiple data, such as multiple data points in a time-series measurement of gene expressions. The Join Brite tool combines a binary relation file and a Brite hierarchy file by an operation similar to the join operation in the relational database. For example, the drug-target

Table 4
KEGG mapper tools

Tool	Query data	Reference knowledge
Search Pathway	kid (K number, org:gene, C number, etc.)	KEGG PATHWAY database
Search Brite	kid (K number, org:gene, C number, etc.)	KEGG BRITE database
Search Module	kid (K number, org:gene, C number, etc.)	KEGG MODULE database
Search&Color Pathway	kid – attribute (color) relations	KEGG PATHWAY database
Search&Color Brite	kid – attribute (color) relations	KEGG BRITE database
Search&Color Module	kid – attribute (color) relations	KEGG MODULE database
Color Pathway	kid – attribute (color, numerical value) relations	Single KEGG pathway map
Color Pathway WebGL	kid – attribute (numerical value) relations	Single KEGG pathway map
Join Brite	kid – attribute (text) relations	Single BRITE hierarchy
Reconstruct Pathway	user-defined id – K number	KEGG PATHWAY database
Reconstruct Brite	user-defined id – K number	KEGG BRITE database
Reconstruct Module	user-defined id – K number	KEGG MODULE database

kid KEGG object identifier (*see* Table 2)

relationships in the KEGG DRUG database can be viewed in a well-organized manner by mapping to the ATC drug classification (www.kegg.jp/brite/br08303_target).

The Reconstruct tools in the fourth type (Reconstruct Pathway, Reconstruct Brite and Reconstruct Module) have been developed especially for genome annotation and comparison. In order to make the tools compatible with the BlastKOALA and KAAS output files, the first column contains user-defined gene identifiers, which are not actually used in the mapping process. Only the K numbers in the second column are used. The mapping result is presented with green coloring used in organism-specific pathway maps. The BlastKOALA service mentioned above contain direct links to these Reconstruct tools. Alternatively, the K number list of the BlastKOALA result file may be stored locally and uploaded through the KEGG Mapper interface. The Reconstruct Pathway and Reconstruct Brite tools can be used for multiple genomes, such as comparing pathways of different species or combining human and gut-microbiome pathways. The Reconstruct Module

tool reports not only complete modules but also incomplete modules with a few missing genes, which may help improve genome annotations.

4 Notes

1. The pathway of phenylpropanoid biosynthesis contains a characteristic reaction sequence pattern of hydroxylation by cytochrome P450 followed by O-methylation, which is defined as the reaction module RM027 (www.kegg.jp/kegg/reaction/rm027.html). While KEGG modules are functional units of genes defined by KO identifiers (K numbers), reaction modules are chemical units obtained by purely chemical analysis without using any gene/protein data and defined by reaction class identifiers (RC numbers) [5]. The reaction class represents a group of reactions with the same local structure transformation patterns and accommodates global structural differences of metabolites.
2. It is interesting to note that there are cases where reaction modules and KEGG modules coincide on the metabolic pathways suggesting coevolution of chemical and genomic aspects of metabolic networks [6]. A case in point is the reaction module RM001 for 2-oxocarboxylic acid chain extension. There are, at least, five known KEGG modules that correspond to this reaction module as summarized in the Brite hierarchy file ko00003 (www.kegg.jp/brite/ko00003). Furthermore, paralogous genes are found for reactions with the same reaction class in different KEGG modules. These observations suggest that reaction modules have a potential to characterize unknown paralogs, which is especially useful in analyzing plant genomes with many paralogs and plant metabolism with many diversified pathways. As indicated in the pathway map of 2-oxocarboxylic acid metabolism (www.kegg.jp/pathway/map01210) there are many unknown genes for the reaction module RM030 for glucosinolate synthesis, which may hopefully be identified by predictive annotation using the empirical relationships between reaction modules and KEGG modules.
3. The following tools at GenomeNet may be of interest for chemical analysis of small molecules and reactions: SIMCOMP (www.genome.jp/tools/simcomp/) for chemical structure similarity search [8] and PathPred (www.genome.jp/tools/pathpred/) for prediction of microbial biodegradation pathways and plant second metabolite biosynthesis pathways [9].

References

1. Kanehisa M, Goto S, Sato Y, Kawashima M, Furumichi M, Tanabe M (2014) Data, information, knowledge and principle: back to metabolism in KEGG. *Nucleic Acids Res* 42:D199–D205
2. Kanehisa M, Goto S (2000) KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Res* 28:27–30
3. UniProt Consortium (2014) Activities at the Universal Protein Resource (UniProt). *Nucleic Acids Res* 42:D191–D198
4. Moriya Y, Itoh M, Okuda S, Yoshizawa A, Kanehisa M (2007) KAAS: an automatic genome annotation and pathway reconstruction server. *Nucleic Acids Res* 35:W182–W185
5. Muto A, Kotera M, Tokimatsu T, Nakagawa Z, Goto S, Kanehisa M (2013) Modular architecture of metabolic pathways revealed by conserved sequences of reactions. *J Chem Inf Model* 53:613–622
6. Kanehisa M (2013) Chemical and genomic evolution of enzyme-catalyzed reaction networks. *FEBS Lett* 587:2731–2737
7. McDonald AG, Boyce S, Tipton KF (2009) ExplorEnz: the primary source of the IUBMB enzyme list. *Nucleic Acids Res* 37:D593–D597
8. Hattori M, Tanaka N, Kanehisa M, Goto S (2010) SIMCOMP/SUBCOMP: chemical structure search servers for network analyses. *Nucleic Acids Res* 38:W652–W656
9. Moriya Y, Shigemizu D, Hattori M, Tokimatsu T, Kotera M, Goto S, Kanehisa M (2010) PathPred: an enzyme-catalyzed metabolic pathway prediction server. *Nucleic Acids Res* 38:W138–W143

Chapter 4

Plant Pathway Databases

Pankaj Jaiswal and Björn Usadel

Abstract

Pathway databases provide information about the role of chemicals, genes, and gene products in the form of protein or RNA, their interactions leading to the formulation of metabolic, transport, regulatory, and signaling reactions. The reactions can then be tethered by the principle of inputs and outputs of one or more reaction to create pathways. This chapter provides a list of various online databases that carry information about plant pathways and provides a brief overview of how to use the pathway databases such as WikiPathways Plants Portal, MapMan and the cereal crop pathway databases like RiceCyc and MaizeCyc, that were developed using the Pathway Tools software.

Key words Biological pathways, Plant pathwaydatabases, Molecular interactions, Metabolic Pathways, Signaling pathways, Regulatory pathways, Transport pathways, WikiPathways, MapMan, RiceCyc, MaizeCyc, Gramene database, Comparative pathway analysis

1 Introduction

After the genomes and transcriptomes are sequenced, researchers are often looking for ways to analyze their genomics datasets to address the biological questions under investigation. Most common analyses include gene function, gene–gene interaction, and pathway enrichment. Many large and small scale studies have focused on identifying genetic and molecular interaction networks involving development, regulatory, signaling, and metabolic pathways. Plant databases like Gramene (<http://www.gramene.org/>), KEGG (Kyoto Encyclopedia for Genes and Genomes; <http://www.genome.jp/kegg/>); Plant Reactome (<http://plantreactome.gramene.org>), MapMan (<http://mapman.gabipd.org/>), MetaCyc (<http://www.metacyc.org>), Plant Metabolic Network (<http://plantcyc.org>), and BioCyc (<http://www.biocyc.org>) are excellent integrated resources for studying the metabolic pathways for plants (Table 1). Similarly much of the regulatory gene–gene interaction networks are stored in databases of molecular interactions such as BAR (<http://bar.utoronto.ca>), ARANet ([David Edwards \(ed.\), *Plant Bioinformatics: Methods and Protocols*, Methods in Molecular Biology, vol. 1374, DOI 10.1007/978-1-4939-3167-5_4, © Springer Science+Business Media New York 2016](http://www.functionalnet.</p></div><div data-bbox=)

Table 1
Plant Pathway database resources

Pathway database name	Species	Source	Reference
RiceCyc	<i>Oryza sativa</i> (rice)	http://pathway.ipiantcollaborative.org/	[1]
MaizeCyc	<i>Zea mays</i> (maize, corn)	http://pathway.ipiantcollaborative.org/	[2]
BrachyCyc	<i>Brachypodium distachyon</i>	http://pathway.ipiantcollaborative.org/	[14, 15]
SorghumCyc	<i>Sorghum bicolor</i>	http://pathway.ipiantcollaborative.org/	[14, 15]
MedicCyc	<i>Medicago truncatula</i>	http://pathway.ipiantcollaborative.org/	[16]
FragariaCyc	<i>Fragaria vesca</i> (strawberry)	Genome database for Rosaceae (GDR) http://pathways.rosaceae.org/ http://pathways.cgrb.oregonstate.edu/	Naithani et al. (unpublished)
VitisCyc	<i>Vitis vinifera</i> (grape)	http://pathways.cgrb.oregonstate.edu/	Naithani et al. (unpublished)
AppleCyc	<i>Malus x domestica</i> (apple)	Genome database for Rosaceae (GDR) http://pathways.rosaceae.org/	Bombarely et al. (unpublished)
PeachCyc	<i>Prunus persica</i> (peach)	Genome database for Rosaceae (GDR) http://pathways.rosaceae.org/	Bombarely et al. (unpublished)

Plant Metabolic Network (PMN)	<p><i>Arabidopsis thaliana</i> col (Arabidopsis) <i>Hordeum vulgare</i> (barley) <i>Brachypodium distachyon</i> (Brachypodium) <i>Manihot esculenta esculenta</i> (cassava, yucca) <i>Brassica rapa</i> ssp. <i>pekinensis</i> (Chinese cabbage) <i>Chlamydomonas reinhardtii</i> (a green alga) <i>Zea mays mays</i> (corn, maize) <i>Vitis vinifera</i> (wine grape) <i>Physcomitrella patens</i> (moss) <i>Oryza sativa japonica</i> group (rice) <i>Carica papaya</i> (papaya) <i>Populus trichocarpa</i> (poplar) <i>Selaginella moellendorffii</i> <i>Setaria italica</i> (Setaria) <i>Sorghum bicolor</i> (sorghum) <i>Glycine max</i> (soybean) <i>Panicum virgatum</i> (switchgrass)</p>	http://www.plantcyc.org/	[17]
KEGG Pathways	Several plant species	Kyoto Encyclopedia of Genes and Genomes	[18, 19]
Plant Reactome	<p><i>O. sativa</i> (rice) <i>A. thaliana</i> <i>Z. mays</i> (~20 more species upcoming)</p>	<p>Gramene database http://plantreactome.gramene.org</p>	[20]
MapMan	Several plant species	<p>MapMan http://mapman.gabipd.org/</p>	[3, 7, 21]
BAR	Several plant species	http://bar.utoronto.ca/	[22]
IntAct		http://www.ebi.ac.uk/intact/	[23]
BIND		http://bind.ca	[24]

[org/aranet/](http://www.functionalnet.org/ricenet/)), RiceNet (<http://www.functionalnet.org/ricenet/>) IntAct (<http://www.ebi.ac.uk/intact/>) and BIND (<http://bind.ca>), to name a few (Table 1).

All of these resources try their best to integrate the most current information. The process involves manual and computational curation efforts, continuous monitoring of new publications, and providing updates on additional interactors. However, despite their best efforts, no single resource is complete with all the desired content and may not provide all the tools necessary to analyze their data. Therefore, in this chapter, we would like to highlight three of the widely used plant pathway platforms, namely, the Pathway-Tools based plant metabolic pathway/genome database PGDBs (for example, RiceCyc [1] and MaizeCyc [2]), the MapMan [3] and the WikiPathways [4, 5]. All three provide unique features and can be used in online and offline (desktop) mode. PGDBs and MapMan are developed using a mix of computational and manual curation and may allow users to edit the content. WikiPathways, on the other hand, with very little centralized content management oversight, allows full control over the data to researchers including editing existing and adding new pathway content. All three platforms are described in the following sections, to help users guide through their workflow. Since these resources and tools undergo frequent updates, we encourage researchers to follow the most updated help documents provided by these resources to keep up with the latest developments.

2 Materials

2.1 Pathway Tools-Based Pathway/ GenomeDatabase (PGDB)

The plant PGDBs provide a platform to search and browse the pathways assembled from metabolic and transport reactions. These reactions are built on the principles of linking inputs and outputs in the form of gene products (catalysts/transporters) and small molecules (compounds). Every instance of the compound is saved in the database only once, thus making sure that the same compound (often associated with different subcellular location) is used in multiple reactions, either as a precursor or derivative. This allows connecting various reactions and pathways to create a network to study and follow the fate of the molecule in a cellular system. Plants, on the other hand, have also undergone, multiple evolutionary events in nature or due to man-made experiments of plant breeding, leading to genome duplication. This leads to the availability of multiple gene copies in the genome either as (homologs) paralogs or homeologs. Often these homologs, due to their phylogenetic relationship, continue to preserve their core function, such as catalyzing the same biochemical reaction, leading to multiple homologs assigned to the same reaction. Alternatively, similar to the single instance of the compound and its associations to

reactions, often gene products (peptide) are mapped to multiple pathways and reactions shown by the same catalyst properties (same reaction but different pathways or same reaction, but different subcellular location). Such association building process provides researchers a mechanism to study the role of genes and gene products, their function and expression that may be regulated spatially or temporally as a result of evolutionary implications on plant's, adaptation, development, and response to growth environment.

Because plants have undergone different kinds of environment and man-made adaptations, inter and intra-specific comparisons reveal subtle differences in collection of genes, genetic variation, pathways, reactions, catalysts, quality and quantity of small molecules. Therefore, it is not advisable to study the pathways and networks from a single reference point of view as we often find in published research articles. PGDB platform allows creation of species/strain-specific genome-scale pathway and small molecule databases to address the specific nuances of the biology. A good number of plant species specific PGDBs have been developed using the same software platform for plants (Table 1) and other organisms [6] (*see Note 1*).

The species-specific PGDB provide options for pathway search, browse, and visualization (Fig. 1), tools for pathway-based analyses of functional genomics datasets, such as those generated in transcriptome, proteome, and metabolome studies. If the user has access to the online portal (Table 1) or the local desktop version hosting multiple species-specific PGDBs, the tools allow species specific comparison as well. For long-term availability, many of the plant PGDBs are available from the iPlant Collaborative (iPlant) infrastructure (Table 1) in collaboration with the Gramene project (*see Note 1*).

2.2 MapMan

MapMan [3, 7] represents a platform specifically dedicated to plant pathways and processes (Fig. 2). However, unlike the Wikipathways platform (described in Subheading 2.3), there are centralized curators to improve the general consistency of the pathways and to centrally annotate genes and metabolites into a specialized ontology allowing these to be painted onto pathway diagrams. Recently, MapMan has merged with the German Plant Primary Database thus guaranteeing its long term availability. Whilst the manual curation effort in MapMan focuses on reference species with a well annotated genome, MapMan supports several dozen plant species and a recently developed extension called Mercator [8] allows for the automatic classification of whole plant transcriptomes and/or proteomes to be classified into the MapMan pathway scheme. MapMan uses more than 50 hand drawn pathway diagrams (*see Note 2*), and features more than 2000 ontology terms, many of which represent individual enzymes in order to make visualizations

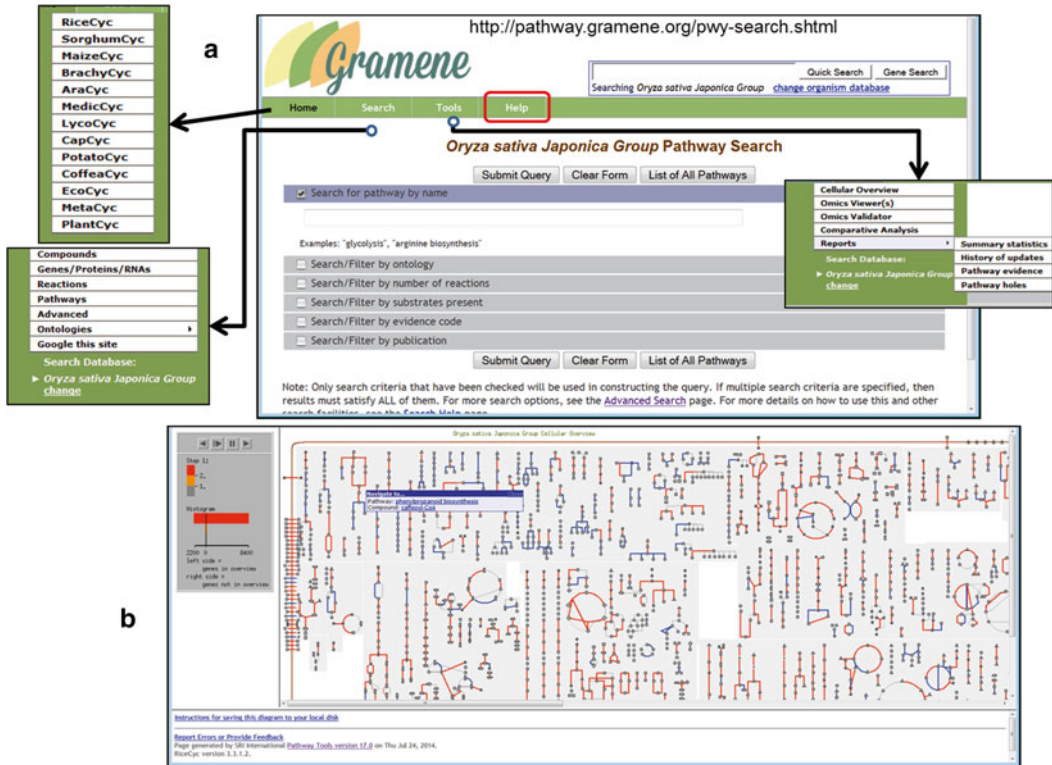


Fig. 1 (a) Screenshot of the Pathway tools-based pathway database (PGDB) search interface provided by the Gramene database. Since each species has its own pathway database, users are required to first select the species of interest from the *top right hand side option* next to the quick search box or from the drop-down menu they would see by hovering the mouse cursor on the ‘Search’ option in the navigation bar. Once the species is selected, there are several options to search any one item and/or combine several searches to create their own filters. The home link on the navigation bar provides direct access to species specific pathway databases from a drop-down menu. Similarly the ‘Tools’ section on the navigation bar provides options to visualize and analyze data, including the OMICs viewer. (b) A screenshot of the rice cellular overview painted with the expression data example described in Subheading 3.1. On the *left* a summary of the uploaded data and color keys are shown and on the main panel the pathways are grouped by their categories, like biosynthetic, catabolic, transport (placed on the periphery of the cellular overview), etc. Hovering the mouse cursor over a pathway depicted by nodes connected by colored lines provides brief information about the pathway and hyperlinks to pathway detail page. In this diagram *red* color means upregulated and *gray color lines* mean down regulated. *Blue colored lines* did not have data mapped to the reactions

more appealing for biologists and to aid the user to quickly grasp the pathway by relying on drawing pathway diagrams similarly as in textbooks. Whilst focusing on the visual representation of the data comprising not only metabolic pathways, but also on general biological processes such as, RNA transcription, protein degradation, and signaling. Many metabolic enzymes and reactions have been linked already which can be queried online at <http://mapman.gabipd.org/web/guest/mapcave>. In addition, data for several species curated by the MapMan collaborator site managed by the researchers in Slovenia, can be explored at www.gomapman.org.

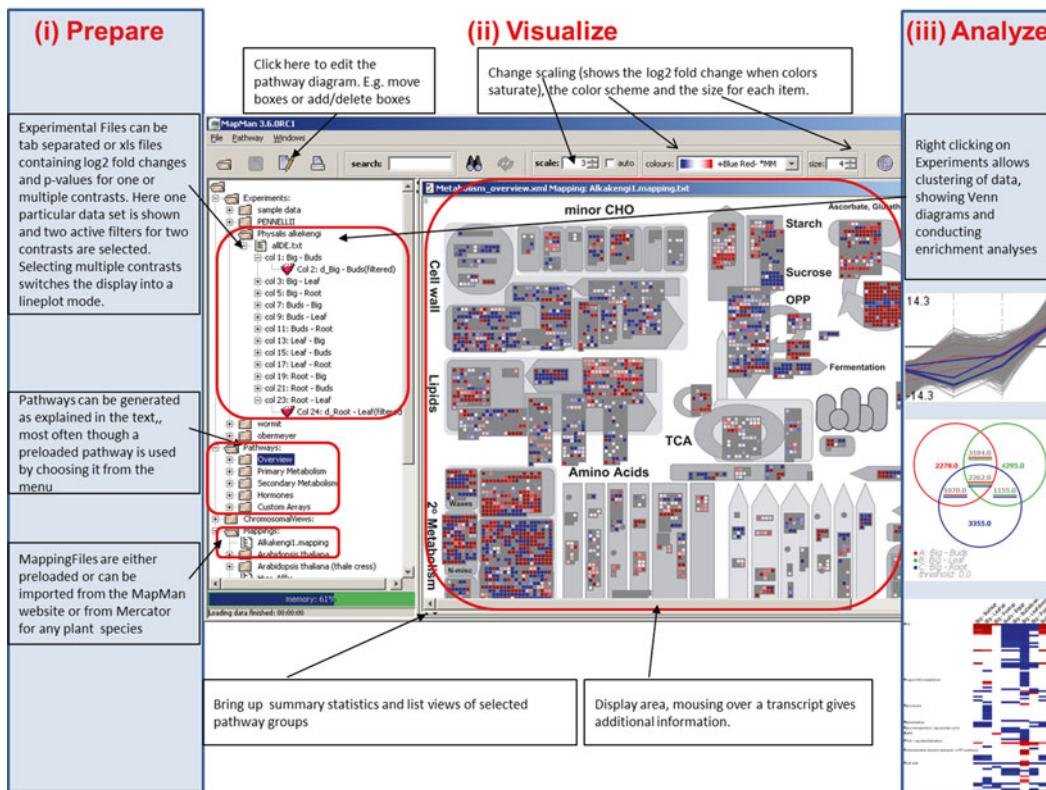


Fig. 2 Using MapMan comprises three stages (a) Data preparation where the user loads the experimental data file, and a mapping file for the plant species of interest. Then the user chooses a pathway to display the data. In the second stage the user can change visualization parameters and use the pathway diagrams to build hypotheses. Finally (c) the user might use clustering and or enrichment tools to get a better idea. This can be interlinked with (b) as for example clusters can be dragged into pathway diagrams to investigate which pathways these comprise

2.3 WikiPathways

With the growing trend of limited funding for dedicated human curators, the onus is now shifting towards developing machine learning techniques (automation) and involving authors of journal publications as expert science freelancers, thus encouraging them to curate the information generated in their publications and maintain the updates by reviewing additional publications. Therefore, by direct involvement of the research community in data curation, the resources are aiming to improve data quality and content (*see Note 3*). In order to achieve this goal, the current plant pathway databases need to develop best practices of curation and the curation tools similar to the Wikipedia style platform. The tool should be conveniently available to users for editing. One such resource for curating pathways and their analysis is the WikiPathways (<http://www.wikipathways.org>), which is also ideal for curation and maintenance of plant pathways [4] by the Plant Biology

research community. WikiPathways is a freely available online portal that allows browsing and statistical data analysis by anonymous users. The registered members of the research community are encouraged to participate in the data quality administration and curation of new and existing pathways (*see Note 3*). The process involves regular updates, and the tool provides ease of editing and data management. WikiPathways, developed as a community curation portal, currently hosts more than 1900 pathways from 25 species. These include several pathways on its Plants Portal (<http://wikipathways.org/index.php/Portal:Plants>) for Arabidopsis, rice, and maize. This portal is a useful resource for the plant biology research community in providing a central platform for pathway curation (Figs. 3 and 4).

Figure 3 consists of four panels labeled (a) through (d). Panel (a) shows the 'Portal:Plants' page on WikiPathways, featuring a navigation menu on the left, a main content area with a 'Log in or create account' button in the top right, and a 'Browse Plant Pathways listed on WikiPathways' section. Panel (b) displays a detailed pathway diagram for 'Vitamin B3 (niacin), NAD and NADP biosynthesis pathway (Zea mays)'. Panel (c) is a zoomed-in view of a specific part of the pathway diagram. Panel (d) shows the 'Gene: GRMZM2G106119' entry from the Gene Ontology database, including a table of GO terms and their associated evidence.

GO ID	Accession	Term	Evidence	Association Source	GOID:GO Term
GO:0005121	GRMZM2G106119	transcript	EA	Ensembl	transcript
GO:0005122	GRMZM2G106119	mRNA	EA	Ensembl	mRNA
GO:0005123	GRMZM2G106119	transcript variant	EA	Ensembl	transcript variant
GO:0005124	GRMZM2G106119	transcript isoform	EA	Ensembl	transcript isoform
GO:0005125	GRMZM2G106119	transcript variant	EA	Ensembl	transcript variant
GO:0005126	GRMZM2G106119	transcript isoform	EA	Ensembl	transcript isoform
GO:0005127	GRMZM2G106119	transcript variant	EA	Ensembl	transcript variant
GO:0005128	GRMZM2G106119	transcript isoform	EA	Ensembl	transcript isoform
GO:0005129	GRMZM2G106119	transcript variant	EA	Ensembl	transcript variant
GO:0005130	GRMZM2G106119	transcript isoform	EA	Ensembl	transcript isoform

Fig. 3 (a) A screenshot of the Plants Portal of the WikiPathways. Users can go to the website www.wikipathways.org and (i) click the Plants Portal link to visit the portal, (ii) browse the species specific pathways, and (iii) register/login to start editing the content. (b) A screenshot of the curated maize vitamin B3 (niacin), NAD, and NADP biosynthesis featured pathway with links for registered users to have access for editing and (iv) options to download in GPML, (v) references to similar information from other database resources, and (vi) download options in standard pathway data exchange formats such as the SBML, Biopax level-3, gene list in text format and for downloading the high resolution image in SVG and other formats (Fig Reproduced with permission from GARNet <http://www.garnetcommunity.org.uk/newsletters>)

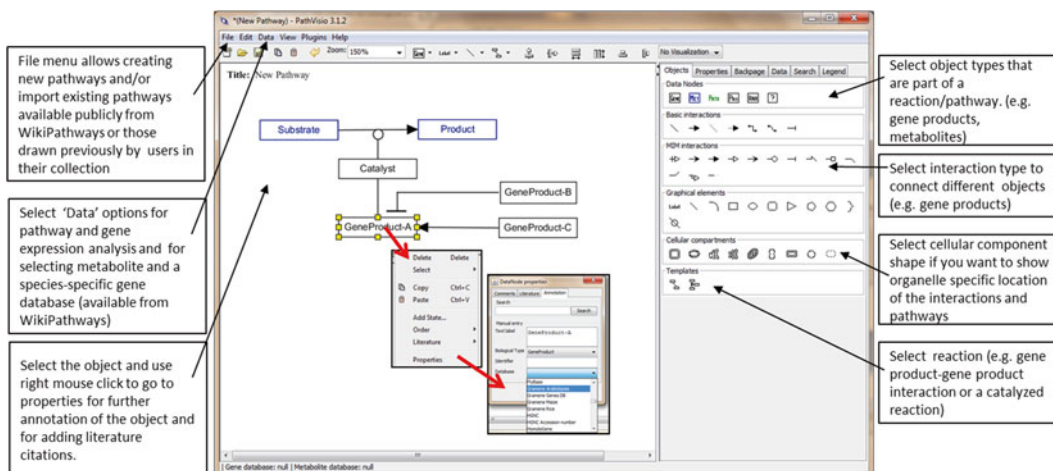


Fig. 4 A screenshot of the desktop pathway editor and analysis tool PathVisio. A similar editor is also provided by the WikiPathways portal for online editing (Reproduced with permission from GARNET <http://www.garnet-community.org.uk/newsletters>)

3 Method

3.1 Pathway Tools-Based Pathway Databases

As mentioned above PGDBs provide basic functionalities of search, browse, and visualize the pathways, reactions, and compounds (Fig. 1). These functionalities have been described in detail in the RiceCyc [1] and MaizeCyc [2] publications (*see Note 1*). For most updated features, please follow the help documents. In the following section, we describe how to use the OMICs viewer tool.

For analyzing the gene expression, metabolite and/or proteomics data, a user needs to have a datafile similar to the file available in supplementary Table 3 <http://www.thericejournal.com/content/6/1/15/additional> [1]. The expression data file needs to be opened in software like MS Excel and saved as a tab-delimited text file, then go to the expression analyses OMICs viewer tool available from any online PGDB webserver (Table 1) such as those provided by Gramene <http://pathway.gemene.org/expression.html>. Follow the instructions listed in sequential order as follows. Also review the details provided in the yellow highlighted section of the OMICs Viewer tool page available from the above listed URL (*see Note 1*).

- Select the species from the top right hand corner. In this example select '*Oryza sativa japonica* group'.
- Select/browse your data file with expression data. For example the tab delimited data provided the supplementary Table 3 <http://www.thericejournal.com/content/6/1/15/additional> [1] (*see Note 1*).

- Select ABSOLUTE/RELATIVE. In this example ‘ABSOLUTE’
- Select if you want relative data from two columns or treating absolute data from each column. In this example, select ‘a single data column’.
- Select Data value: select 0-centered/1-centered scale. In this example 1-centered scale.
- Data columns (numerators in ratios): type the column number(s) which have expression data. Remember the tool interprets the column #1 in your file as column#0 which contains the gene/compound_ID/name/synonym. Ideally your data starts from column #1 onwards (same as column #2 in your file). In this time series example, let’s say if we are interested in analyzing the first four time points, type digits from 1 to 4 (one digit per row) by identifying your first four data columns.
- If users want to do a relative data analysis, one is allowed to identify the denominator data column after listing primary numerator data column(s)
- Choose a color scheme of your choice by selecting either, full spectrum, computed from data provided (default) and define your own cutoff/thresholds. In this example, select the three color scheme with a threshold of 5.
- Select which type of overview you would like to see painted with your data. For example, a cellular overview (default) of all the pathways and reactions, genome/chromosomal view, or simply generate a table view with your suggested threshold. In this example, select the default cellular overview.
- Hit the ‘Submit’ button. It would generate a cellular overview and display the expression pattern of genes mapped to individual reactions in the previously selected color scheme (Fig. 1) (*Note*: a large data set may take a longer time to load the data and return results).
- Once you see the returned results, by hovering your cursor over the reaction and the pathway a user can find hyperlinks to detail information page.
- Independent of the source of the online portal (Table 1), if the plant PGDB was generated using the Ptools software, the OMICs viewer tool would work exactly the same way, since all these portals run the same software and the database to serve their respective pathway datasets.

3.2 MapMan

When users want to use MapMan to visualize expression information, metabolite and or protein accumulation on pathway data, all they need to do is to download the MapMan package. This comprises the actual MapMan software written in Java that runs on any

platform, pathway diagrams as well as accompanying “mapping files” which classify individual genes into the MapMan ontology. In the cases where the species of interest is not supported by MapMan, users can generate an automated custom annotation based on the Mercator tool [8] by submitting FASTA formatted sequences online. After having loaded the provided or custom generated 'mapping file', the users can benefit from the supplied pathway diagrams by loading their own datasets in Excel or text format. The file to be loaded into MapMan needs to be formatted in such a way that each line in the file begins with the identifier of an analyte (transcript, metabolite, etc.) and the following columns give numerical values for the behavior of the analyte in different conditions. In each case the numerical value should be log transformed, e.g., by referring it to a reference or simply by providing log fold changes. To give an example a typical line could be comprised of At1g53500, -2.0, 1.0, 2.0, telling MapMan that the Arabidopsis gene At1g53500 is downregulated fourfold in the first condition, upregulated twofold in the second condition and upregulated fourfold in the last condition, assuming log₂ was used. In order to identify these conditions the first line should contain column headings, but this is not strictly necessary in which case MapMan will simply treat these conditions as anonymous. The data are then mapped onto the pathway where each analyte is represented by symbols which are color coded according to the log fold change supplied in the experimental data file supplied by the users. To visualize time courses, the user only needs to select multiple time points from a set making MapMan change into a line plot display.

Since, MapMan is ontology-based, it is thus also possible to test for an enrichment of upregulated or downregulated genes by invoking the built-in enrichment testing module [9].

In cases where the user wants to improve or change the annotation of individual genes, the underlying “mapping file” can simply be opened in Microsoft Excel, Libre Office or in a text based editor and changed there (*see Note 2*). The “mapping file” simply describes one association per line, comprising the MapMan “Bincode”, the human readable name of this ontological item, the gene/metabolite/protein name, the description of this item to be displayed in MapMan and finally the data type, usually T for transcripts or M for metabolites. Again an example might be 10.1.10, cell wall.precursor synthesis.RHM, At1g53500, mum4 or rhm2 involved in UDP-rha synthesis, T. This tells MapMan that the gene At1g53500 is to be placed into the Bincode 10.1.10, whose human readable description is cell wall.precursor synthesis.RHM, whereas the gene description would be “mum4 or rhm2 involved in UDP-rha synthesis” and that this is actually a transcript (T), thus MapMan will visualize this using a square.

If the user wants to generate their own pathway, they simply have to load a graphical representation of the pathway diagram to be displayed which can be generated in any typical image processing programs or in Microsoft PowerPoint. Whilst bitmap formats such as png or jpeg are supported, it is advisable to save the pathway diagram in svg format as this allows lossless scaling of the pathway diagram and export in publication ready quality of 300 dpi or more (*see Note 2*). After import into MapMan the diagram is annotated by clicking on positions on the pathway and adding a particular MapMan Bincode at a particular position relying on the MapMan ontology read from the “mapping file” (*see above*). Once this is done the pathway can be saved and/or even distributed via the MapMan website by contacting the website administrator or by using the MapMan forum. Since, MapMan does not directly map genes or metabolites, but uses the ontology terms representing genes or metabolites; therefore, each pathway generated using the canonical ontology can thus be used for any plant species.

In cases where an additional pathway necessitates an extension of the ontology, the user can simply do so in the “mapping file” or even devise a completely new ontology specific for a pathway of interest. However an extension of the existing ontology where enzymes from one pathway are usually grouped together is advisable, as in such a case the resulting data can be made available to the community and be incorporated into future MapMan releases.

3.3 WikiPathways

For the users interested in curation, the actual networks drawn using the online version of the pathway editor provided by the WikiPathways and their stand alone Java application PathVisio (Fig. 4) [10], involves simple network data standards that allow drawing nodes (gene products, catalysts, metabolites, etc.) and edges (interaction types) connecting the nodes that can be extracted in standardized network exchange formats such as BioPax, SBML, and *Simple Interaction Format* (SIF). PathVisio the standalone pathway editor can be downloaded from the website <http://www.pathvisio.org> and installed locally on the desktop. Following the instructions provided in the help documents and a set of tutorials (<http://www.pathvisio.org/wiki/PathVisioTutorials>), data nodes (genes and metabolites) and interactions (edges/connectors) between two nodes can be drawn (*see Note 3*). The interaction arrows represent activation/upregulation, and T-bars represent inhibition/downregulation. A set of genes with the same function (for example the paralogs or isozymes) can be grouped together. Similarly, genes encoding for the subunits of a protein or functional complex can be grouped as complex, where necessary. Each gene node is labeled with gene symbol and the metabolite one with the name of the metabolite. Each node, depending on its type (gene/metabolite) wherever possible must have reference database IDs, e.g., gene IDs refer to the Gramene and Ensembl Gene IDs for plants species

Arabidopsis/rice/maize/etc. Metabolites are referred to either CAS numbers or ChEBI IDs depending on their availability in these resources. Additional One or more references to PubMed literature IDs and any useful comments in free text format can be added to either or all of nodes, edges or the pathway. Despite the use of community standards in the curation of these networks, they do not limit the community curators and authors from curating their own style, for example, nodes and edges/interactions can be color-coded to reflect functional classification such as external stimulus, subcellular localization, and self-interactions. Considering that some of the pathways drawn by the individuals may not be public, users are allowed to create and save their own pathway drawing and use them for gene expression analysis by using the desktop version of PathVisio available from <http://www.pathvisio.org>. As and when the time permits, registered users of WikiPathways can upload their pathway diagram data saved by the desktop version of PathVisio in the GPML format to the online portal and choose to keep it private, share it with other registered users and/or make it public (*see* **Note 3**). More recent updates and help can be sought by following the PathVisio tutorial (<http://www.pathvisio.org/documentation/tutorials/tutorial-1/>) and WikiPathways (<http://www.wikipathways.org/index.php/Help:Tutorial>).

In addition to drawing the pathways the information provided in the diagrams can also be used for pathway enrichment and gene expression analysis. For gene expression analysis, users need to map the expression of genes to the subset chosen in the pathways. This is done by either using the Pathvisio (<http://www.pathvisio.org>) and/or by GenMAPP-CS (<http://www.genmapp.org/beta/genmappcs/>) [10, 11] tools. Tools work by downloading the species specific gene database provided through the BridgeDb [12, 13] framework (<http://www.bridgedb.org>), to visualize and analyze, the expression data stored in a CSV/tab-delimited text file. For more details on how to perform the gene expression analysis follow the help documents and guides from PathVisio (<http://www.pathvisio.org/documentation/tutorials/tutorial-2/>) and GenMAPP-CS (<http://opentutorials.cgl.ucsf.edu/index.php/Tutorial:ExpressionAnalysisGenMAPP-CS>)

4 Notes

1. Pathway Tools

- Even though we have used the examples from Gramene, almost all the online resources listed in Table 1 serving the Pathway Tools based PGDB, work similarly. Differences may include the look and feel of the webpages, besides the methods used in compiling the primary annotations for building the pathway annotations.

- The functionalities presented in this chapter are based on the version 17 of Pathway Tools provided by the SRI International. Different websites providing pathway tools based on this tool may be served using a version older than 17.0 or a much newer version. In the newer version though basic functionalities remain the same, the additional options are provided to paint and visualize the expression data.
- We encourage users to get a licensed copy of the Pathway Tools software from SRI International to run on their local server and desktop version. This desktop version of the software allow users additional features (<http://metacyc.org/desktop-vs-web-mode.shtml>) such as flux balance analysis, interspecific pathway comparison in cellular overview, and metabolite tracing. The local installation also allow users to edit and add the pathways by following the software's help guideline. However, if the users end up editing contents of an existing species-specific database available for local use from various sources (Table 1), we encourage users to submit their edits to the source provider for future integration in the PGDB as part of community contribution.
- The Expression analysis tool does not perform the analysis of raw data. Users would have to run additional third party software to analyze their experimental data. This tool only maps the gene, compound, IDs and names to reactions and paints the expression profile of the genes and compounds for visualizing the expression differences.
- For advanced users, the Pathway-Tools based species-specific online pathway databases, also provide the data via webservices. For more details follow the webservices documentation at <http://biocyc.org/web-services.shtml>. This website provides a generic outline. The URL to access the data objects in ptools-xml format may need customization depending on the online source of the data a user may be interested in querying.

2. MapMan

- While pathways in MapMan can be drawn as described above in MapMan sections, it is currently not possible to directly draw pathway diagrams in MapMan simply due to the fact that it is difficult to reproduce user friendly image manipulation programs such as PowerPoint and Adobe Illustrator. However, in the case that this was of great user interest this functionality would be added in a future version.

- At the moment it is not yet possible to export the database, including reaction information in SBML format.
- MapMan is dual licensed since 2014 (CC-BY-ND and CC-BY-SA-NC) allowing completely free use, reuse, and modification by academic users, provided they cite MapMan and use in the commercial field, however without the possibility to make any changes or extensions.

3. WikiPathways

- While Wikipathways do provide information on the plant pathways it does not have a full or a large number of plant pathways. Therefore, unlike gene expression and pathway enrichment analyses that can be performed on Wikipathways for humans with large number of annotated pathways, the plant dataset is still in its infancy.
- Since Wikipathways allow registered users to create, edit, and add new pathway datasets, it encourages plant biologists to contribute their data/knowledge to help improve the curated pathway data deficiency.
- If a plant species is not already listed, users would have to request its addition to the list and its genome database (via BidgeDB) prior to adding pathways from this new species.
- For advanced users interested in accessing the data programmatically through webservices, please follow the instructions provided at http://www.wikipathways.org/index.php/Help:WikiPathways_Webservice.

Acknowledgement

We kindly acknowledge current and former members of our respective laboratories and collaborative projects for their contribution to building pathway resources for the plant biology community. We also acknowledge the iPlant Collaborative (<http://www.iplantcollaborative.org/>) for hosting the Gramene's PGDB pathway databases (<http://pathway.iplantcollaborative.org/>). We extend special thanks to numerous collaborators and domain experts who continue to contribute to the development and annotation of pathway databases. PJ acknowledges funding support by the US National Science Foundation (NSF) award IOS #1127112 and the funds made available by the Oregon State University. BU acknowledges funding support for the PlabiPD project FKZ 0315961

References

- Dharmawardhana P, Ren L, Amarasinghe V, Monaco M, Thomason J, Ravenscroft D, McCouch S, Ware D, Jaiswal P (2013) A genome scale metabolic network for rice and accompanying analysis of tryptophan, auxin and serotonin biosynthesis regulation under biotic stress. *Rice (N Y)* 6(1):15
- Monaco M, Sen TZ, Dharmawardhana P, Ren L, Schaeffer M, Naithani S, Amarasinghe V, Thomason J, Harper L, Gardiner J, Cannon E, Lawrence C, Ware D, Jaiswal P (2013) Maize metabolic network construction and transcriptome analysis. *Plant Genome* 6:1–12
- Usadel B, Poree F, Nagel A, Lohse M, Czedik-Eysenberg A, Stitt M (2009) A guide to using MapMan to visualize and compare Omics data in plants: a case study in the crop species, Maize. *Plant Cell Environ* 32(9):1211–1229
- Hanumappa M, Preece J, Elser J, Nemeth D, Bono G, Wu K, Jaiswal P (2013) WikiPathways for plants: a community pathway curation portal and a case study in rice and Arabidopsis seed development networks. *Rice (N Y)* 6(1):14
- Kelder T, van Iersel MP, Hanspers K, Kutmon M, Conklin BR, Evelo CT, Pico AR (2011) WikiPathways: building research communities on biological pathways. *Nucleic Acids Res* 40(Database issue):D1301–D1307
- Caspi R, Altman T, Billington R, Dreher K, Foerster H, Fulcher CA, Holland TA, Keseler IM, Kothari A, Kubo A, Krummenacker M, Latendresse M, Mueller LA, Ong Q, Paley S, Subhraveti P, Weaver DS, Weerasinghe D, Zhang P, Karp PD (2014) The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases. *Nucleic Acids Res* 42(Database issue):D459–D471
- Thimm BO, Gibon Y, Nagel A, Meyer S, Kruger P, Selbig J, Muller LA, Rhee SY, Stitt M (2004) MAPMAN: a user-driven tool to display genomics data sets onto diagrams of metabolic pathways and other biological processes. *Plant J* 37(6):914–939
- Lohse M, Nagel A, Herter T, May P, Schroda M, Zrenner R, Tohge T, Fernie AR, Stitt M, Usadel B (2014) Mercator: a fast and simple web server for genome scale functional annotation of plant sequence data. *Plant Cell Environ* 37(5):1250–1258
- Usadel B, Nagel A, Steinhauser D, Gibon Y, Blasing OE, Redestig H, Sreenivasulu N, Krall L, Hannah MA, Poree F, Fernie AR, Stitt M (2006) PageMan: an interactive ontology tool to generate, display, and annotate overview graphs for profiling experiments. *BMC Bioinform* 7:535
- van Iersel MP, Kelder T, Pico AR, Hanspers K, Coort S, Conklin BR, Evelo C (2008) Presenting and exploring biological pathways with PathVisio. *BMC Bioinform* 9:399
- Doniger SW, Salomonis N, Dahlquist KD, Vranizan K, Lawlor SC, Conklin BR (2003) MAPPFinder: using Gene Ontology and GenMAPP to create a global gene-expression profile from microarray data. *Genome Biol* 4(1):6
- Gao J, Zhang C, van Iersel M, Zhang L, Xu D, Schultz N, Pico RA (2014) BridgeDb app: unifying identifier mapping services for Cytoscape. *F1000Res* 3:148
- van Iersel MP, Pico AR, Kelder T, Gao J, Ho I, Hanspers K, Conklin BR, Evelo CT (2010) The BridgeDb framework: standardized access to gene, protein and metabolite identifier mapping services. *BMC Bioinform* 11:5
- Jaiswal P (2011) Gramene database: a hub for comparative plant genomics. *Methods Mol Biol* 678:247–275
- Jaiswal P, Ni J, Yap I, Ware D, Spooner W, Youens-Clark K, Ren L, Liang C, Hurwitz B, Zhao W, Ratnapu K, Faga B, Canaran P, Fogleman M, Hebbard C, Avraham S, Schmidt S, Casstevens TM, Buckler ES, Stein L, McCouch S (2006) Gramene: a genomics and genetics resource for rice. *Rice Genet Newslett* 22(1):9–16
- Urbanczyk-Wochniak E, Sumner LW (2007) MedicCyc: a biochemical pathway database for *Medicago truncatula*. *Bioinformatics* 23(11):1418–1423
- Zhang P, Dreher K, Karthikeyan A, Chi A, Pujar A, Caspi R, Karp P, Kirkup V, Latendresse M, Lee C, Mueller LA, Muller R, Rhee SY (2010) Creation of a genome-wide metabolic pathway database for *Populus trichocarpa* using a new approach for reconstruction and curation of metabolic pathways for plants. *Plant Physiol* 153(4):1479–1491
- Nakao M, Bono H, Kawashima S, Kamiya T, Sato K, Goto S, Kanehisa M (1999) Genome-scale gene expression analysis and pathway reconstruction in KEGG. *Genome Inform Ser Workshop Genome Inform* 10:94–103
- Ogata H, Goto S, Fujibuchi W, Kanehisa M (1998) Computation with the KEGG pathway database. *Biosystems* 47(1-2):119–128
- Monaco MK, Stein J, Naithani S, Wei S, Dharmawardhana P, Kumari S, Amarasinghe V, Youens-Clark K, Thomason J, Preece J, Pasternak S, Olson A, Jiao Y, Lu Z, Bolser D, Kerhornou A, Staines D, Walts B, Wu G, D'Eustachio P, Haw R, Croft D, Kersey PJ, Stein L, Jaiswal P, Ware D (2014) Gramene

- 2013: comparative plant genomics resources. *Nucleic Acids Res* 42(Database issue): D1193–D1199
21. Ling MH, Rabara RC, Tripathi P, Rushton PJ, Ge SX (2013) Extending MapMan ontology to tobacco for visualization of gene expression. *Dataset Pap Biol* 2013:pii: 706465
 22. Toufighi K, Brady SM, Austin R, Ly E, Provart NJ (2005) The Botany Array Resource: e-Northern, Expression Angling, and promoter analyses. *Plant J* 43(1):153–163
 23. Orchard S, Ammari M, Aranda B, Breuza L, Briganti L, Broackes-Carter F, Campbell NH, Chavali G, Chen C, del-Toro N, Duesbury M, Dumousseau M, Galeota E, Hinz U, Iannuccelli M, Jagannathan S, Jimenez R, Khadake J, Lagreid A, Licata L, Lovering RC, Meldal B, Melidoni AN, Milagros M, Peluso D, Perfetto L, Porras P, Raghunath A, Ricard-Blum S, Roechert B, Stutz A, Tognolli M, van Roey K, Cesareni G, Hermjakob H (2014) The MIntAct project--IntAct as a common curation platform for 11 molecular interaction databases. *Nucleic Acids Res* 42(Database issue):D358–D363
 24. Isserlin R, El-Badrawi RA, Bader GD (2011) The biomolecular interaction network database in PSI-MI 2.5. *Database (Oxford)* 2011:baq037

The Plant Ontology: A Tool for Plant Genomics

Laurel Cooper and Pankaj Jaiswal

Abstract

The use of controlled, structured vocabularies (ontologies) has become a critical tool for scientists in the post-genomic era of massive datasets. Adoption and integration of common vocabularies and annotation practices enables cross-species comparative analyses and increases data sharing and reusability. The Plant Ontology (PO; <http://www.plantontology.org/>) describes plant anatomy, morphology, and the stages of plant development, and offers a database of plant genomics annotations associated to the PO terms. The scope of the PO has grown from its original design covering only rice, maize, and *Arabidopsis*, and now includes terms to describe all green plants from angiosperms to green algae.

This chapter introduces how the PO and other related ontologies are constructed and organized, including languages and software used for ontology development, and provides an overview of the key features. Detailed instructions illustrate how to search and browse the PO database and access the associated annotation data. Users are encouraged to provide input on the ontology through the online term request form and contribute datasets for integration in the PO database.

Key words Bioinformatics, Ontology, Plant anatomy, Plant development, Comparative genomics, Genomeannotation, Transcriptomics, Phenomics, Semantic web

1 Introduction

All areas of modern science are encountering the problem of data overload, and plant science is no exception. Currently, there are more than 60 sequenced plant genomes in various states of completion [1], and about 40 of those are hosted by Phytozome, the Joint Genome Institute's comparative plant genomics portal (<http://phytozome.jgi.doe.gov/pz/portal.html#>), and the number is growing. In addition, many *de novo* and reference-guided transcriptomes are becoming available for novel and reference plant species [2–7]. The first sequenced plant genome was the model plant *Arabidopsis thaliana*, with a small genome size of 135 Mb, arranged in five chromosomes [8]. Recently sequenced examples of more complex plant genomes are *Eucalyptus grandis* [9], Norway Spruce (*Picea abies*) [10], and the common bean *Phaseolus vulgaris* [11], economically and agronomically important species.

Along with this increasing deluge of genomic and comparative data is a massive amount of transcriptome data on the expression of individual genes and groups of genes. Therefore, the complexity of the genomic datasets is increasing as well.

Efforts are underway to develop means to encourage sharing of these large datasets, such as the emerging format ISA-TAB [12] and data repositories such as those at Gigascience (<http://www.gigasciencejournal.com/>), NCBI's Sequence Read Archive (SRA; <http://www.ncbi.nlm.nih.gov/sra>), the Transcriptome Shotgun Assembly Sequence Database (TSA; <http://www.ncbi.nlm.nih.gov/genbank/tsa>), the Gene Expression Omnibus (GEO; <http://www.ncbi.nlm.nih.gov/geo/>), ArrayExpress (<http://www.ebi.ac.uk/arrayexpress/>), Dryad (<http://datadryad.org/>) and others.

In order to retrieve, analyze, and search the large datasets, mechanisms must be implemented to relate the datasets to one another, and to allow cross-species comparative analyses. Datasets need to be described using a standardized set of metadata with a common vocabulary shared between them. Efforts are underway to define such minimum information standards in many fields of biology [13–15]. Metadata, often described as “data describing the data” [16], provides the information necessary for relating datasets to each other, but one of the challenges of effective metadata use is ensuring that related fields are comparable between experiments [17]. For example, precisely defining the source of a plant tissue sample, a treatment, an experimental condition, or a related gene expression observation is complicated by different terminologies used for various types of experiments and plant species. The use of controlled vocabularies or “ontologies” can mitigate these problems and allow data sharing and integration across various plant genomics resources, and comparisons both within and between species to be made.

1.1 What Is an Ontology?

In the world of computer science and bioinformatics, an ontology is a representation of a body of knowledge that exists about objects in a domain of interest, the categories of those objects, and their inter-relationships. The categories of objects, or “classes,” are a conceptualization of the knowledge that exists about the objects [18]. The terms in the ontology are the agreed-upon labels for the classes and the subclasses, which are the categories of objects described by the ontology. The ontology constitutes a controlled and structured vocabulary of terms to represent the knowledge about the types of entities within a given domain [19, 20]. The use of ontologies has gained increasing importance as the number, complexity, and size of biological datasets have increased [17].

1.2 Background: Development of GO and PO

One of the earliest pioneers in the development and use of ontologies for biology was the Gene Ontology (GO; <http://www.geneontology.org/>), developed in the late 1990s [21, 22]. The GO has become an established standard for describing the functions of genes

and gene products and is utilized in many plant genomic databases and resources such as TAIR (<https://www.arabidopsis.org/>), MaizeGDB (<http://www.maizegdb.org/>) [23–25], and Ensembl Plants (<http://plants.ensembl.org>) [26]. In addition, GO functional gene annotations are commonly presented in plant genome [e.g., [27–29]] and transcriptome [e.g., [2, 30–32]] studies.

Early development of the Plant Ontology (PO) arose out of the efforts of individual model angiosperm research communities to describe the anatomy and growth stages of *Arabidopsis thaliana* [33], maize [34], and rice [35]. As the potential for comparative genomics grew with the increasing amounts of plant genomics data becoming available, the Plant Ontology [36] was developed to provide terminology to describe plant anatomy and morphology [37] and developmental stages [38]. Expansion of the PO to accommodate emerging model plants and needs of the user community has continued for the past several years, and now includes terms to describe anatomy and developmental stages of all green plants [20]. The goal of the PO is to serve as a reference ontology and annotation database for all green plants, from algae to angiosperms.

The purpose of this chapter is to introduce the concept of ontologies and to provide an overview of the status of the Plant Ontology. We will introduce how the PO and related biological ontologies are organized, and how to search and browse the ontology terms and the associated annotation data. In addition, we will demonstrate how to provide input on the ontology terms, and how users can utilize and/or contribute annotated datasets. Throughout the chapter, ontology terms and relations are printed in *italics*. This article describes the PO in reference to Release #20 (August 2013).

2 Materials

2.1 The Plant Ontology (PO)

The PO consists of two interconnected branches (also called aspects), which together describe the morphological and anatomical structures such as *plant organ*, *whole plant*, and *plant cell*, and the stages of development of these plant structures (including the *whole plant*) (Fig. 1). Each branch is organized hierarchically under a single root class: *plant anatomical entity* (PAE) and *plant structure development stage* (PSDS), respectively. Similar to a taxonomic tree, in an ontology graph, the classes found towards the root of the tree are the most general, with increasing granularity as you move farther from the root [20].

2.1.1 The Plant Anatomical Entity (PAE) Branch

The classes in *PAE* branch (Fig. 1a) describe the morphological and anatomical structures of a plant and are defined based on their structure and location, rather than their function [20], following the principles set out in the Foundational Model of Anatomy

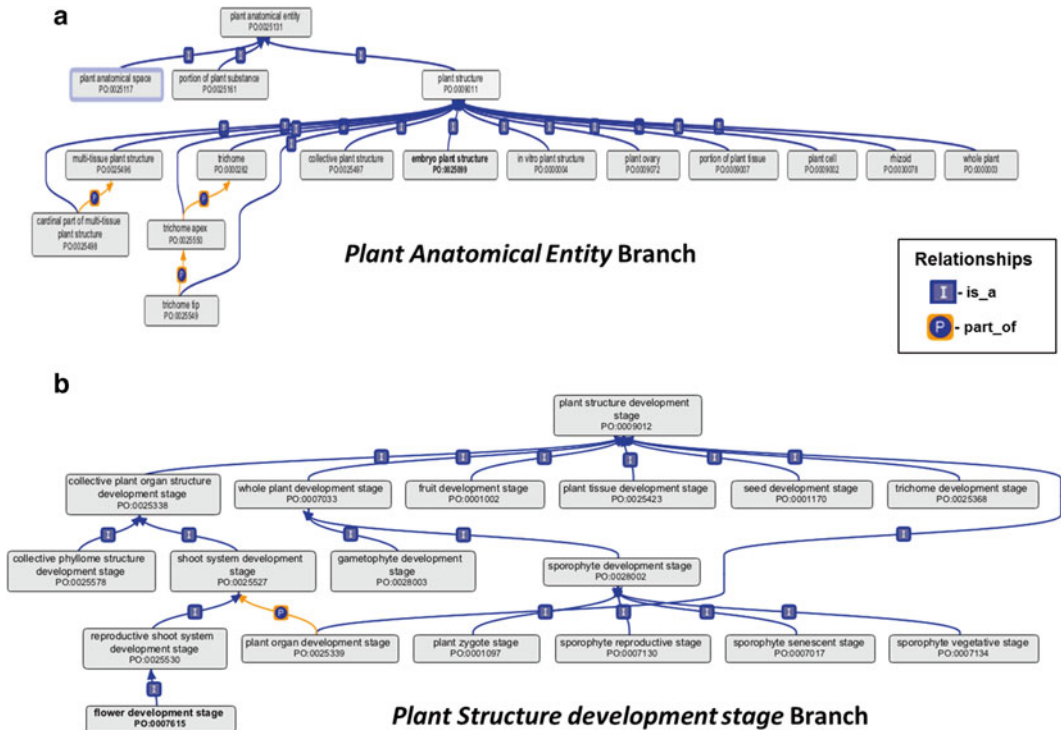


Fig. 1 Upper level structure of the two branches of the Plant Ontology. The Plant Ontology (PO) is made up of two interconnected branches organized hierarchically under a root class *Plant Anatomical Entity* or *Plant structure development stage*. The classes are linked by relationships (read in the direction of the arrows) with the most general classes towards the top. Examples are shown of only the *is_a* and *part_of* relations. This image was based on the PO Version #20 (Released August 2013) and was created using OBO-Edit [49]. (a) The *Plant Anatomical Entity* (PAE) branch has three upper-level subclasses: *plant structure*, *plant anatomical space*, and *portion of plant substance*, which together describe all the anatomical and morphological parts of a plant. The largest subclass *plant structure* has 13 direct subclasses, which each have many subclasses (not shown in figure). (b) The *plant structure development stage* branch describes the developmental stages of all *plant structures*, including the *whole plant*. There are six direct subclasses, which describe the developmental stages of the classes of *plant structures*, including the *whole plant*

(FMA) [39]. For example, the common *plant organ*, the *root*, is defined as: “A *plant axis* (PO:0025004) that lacks *shoot axis nodes* (PO:0005004), grows indeterminately, and is usually positively geotropic,” rather than being defined based on the root functions: absorption of water and inorganic nutrients, anchoring the plant body to the ground, and supporting it, storage of food and nutrients, and vegetative reproduction, which are added as a comment to support the definition. The *plant anatomical entity* branch is further divided into three main subclasses (1) *plant anatomical space*; (2) *portion of plant substance*; and (3) *plant structure*, which are based on existing terms from the Common Anatomy Reference Ontology (CARO) [40].

The subclass *plant anatomical space* has 35 subclasses which describe plant parts such as an *axil*, *locule*, or *stomatal pore* and other spaces that are part of a plant, and surrounded by one or more anatomical structures [20]. This class only includes those spaces generated by developmental, morphogenetic, or other physiological processes, and excludes those such as random spaces between leaves or branches. The other main subclass of the PAE, *portion of plant substance*, has 14 subclasses which describe entities that are produced by the plant, but are not structures themselves such as *plant cuticle*, *cuticular wax*, *plant sap*, and *cutin*.

By far the largest group, the *plant structure* class, with 1212 subclasses (Release #20; as of Aug. 2013) describes all the traditional plant parts, such as *plant organ*, *plant cell*, and *whole plant*, and is grouped into 13 upper-level subclasses (Fig. 1a), including some special classes needed for categorization such as *multi-tissue plant structure* and *collective plant structure*. These terms will not be familiar to plant biologists, but are necessary to provide a complete and inclusive structure and to support interspecific comparisons of plant structures. The vast majority of the common plant structures found in angiosperms, gymnosperms, and pteridophytes are described in the PO, and the ontology has been expanded from its original mandate to also describe the non-vascular plants such as bryophytes and green algae [19].

2.1.2 The Plant Structure Development Stage (PSDS) Branch

The PSDS branch of the PO is organized hierarchically under the root class *plant structure development stage* (Fig. 1b). It describes the stages of development of specific *plant structures*, including those of the *whole plant*, and more or less mirrors the hierarchy of the *plant structure* branch of the PO, with some special classes needed for categorization such as *collective plant organ structure development stage* and *collective phyllome structure development stage*.

The *whole plant developmental stage* class describes the plant life cycles of both vascular and non-vascular plants through the two-phase alteration of generations; the *gametophyte development stage* and the *sporophyte development stage*. In bryophytes, the dominant haploid gametophyte and a minor diploid sporophyte phase [41] are in contrast to the developmental patterns of angiosperms and gymnosperms, where the dominant stage is the *sporophyte development stage*. The *sporophyte development stage* starts with *plant zygote stage*, which begins after GO:*fertilization* (a GO *biological process* term) has occurred, and continues through the *sporophyte senescent stage*, and eventually death of the plant (*see Note 1*). Currently the *plant structure development stage* is under active development and enrichment, and consists of 289 terms as of Release #20 (August 2013).

2.1.3 The PO Annotation Database

In addition to the ontology itself, the annotation database (Table 1) is a key feature of the Plant Ontology, and other ontologies such as the Gene Ontology [22, 42]. The annotations are essentially links

Table 1
List of the sources that provided annotation datasets to the PO database

Source	Plant species	Type of data	# Associations to PO terms
MaizeGDB	<i>Zea mays</i>	genes and gene products; germplasm	1,495,156
TAIR	<i>Arabidopsis thaliana</i>	genes and gene products	532,679
Rensing lab and cosmoss	<i>Physcomitrella patens</i>	genes and gene products	81,875
Jaiswal lab	<i>Oryza sativa</i> (japonica subgroup)	genes and gene products	73,952
Gramene	<i>Oryza sativa</i>	genes and gene products; QTL	29,867
Sol Genomics Network (SGN)	<i>Lycopersicon esculentum</i> (tomato)	genes and gene products, germplasm	18,949
Jaiswal lab	<i>Fragaria vesca</i> (strawberry)	genes and gene products	9561
MaizeGenetics Cooperation Stock Center	<i>Zea mays</i>	germplasm	5070
PO curators	<i>Vitis vinifera</i> (grape)	genes and gene products	3675
NASC (European Arabidopsis Stock Centre)	<i>Arabidopsis thaliana</i>	germplasm	1897
Sol Genomics Network (SGN)	<i>Solanum insanum/Solanum melongena</i> (eggplant)	genes and gene products, germplasm	873
AgBase	<i>Gossypium hirsutum</i> (cotton)	genes and gene products	472
Various	Other species	genes and gene products, germplasm	206
Total:			2,254,233

See Table 3 for links

or “associations” between PO terms and plant genomics data sourced from datasets from around the world. As of Release #20 (Aug. 2013), the PO annotation database contains 142,446 unique data objects representing genes, gene models, proteins, RNAs, germplasm sources, and quantitative trait loci (QTLs), which form about 2.25 million annotations to PO terms, as many of the data objects are annotated to more than one PO class [20]. The annotation datasets come from ten different data sources; primarily from collaborating model organism database groups, although some are added directly by the POC curation team, and some are contributed directly by authors (Table 1). The scope of the data encompasses 22 different plant species, ranging from the moss *Physcomitrella patens* to *Arabidopsis* and *Zea mays*.

2.2 Key Elements of the Plant Ontology and Other Related Bio-ontologies

The Plant Ontology follows the guidelines established by the Open Biological and Biomedical Ontologies (OBO) Foundry initiative (<http://www.obofoundry.org/about.shtml>) [43]. The OBO Foundry is a collaboration among science-based ontology developers that aims to establish a set of best practices for ontology development, with the goal of creating a suite of orthogonal, interoperable reference ontologies in the biomedical domain [43]. In 2013, the PO was accepted as a full member OBO Foundry Ontology (http://www.obofoundry.org/wiki/index.php/Results_of_OBO_Foundry_Reviews).

2.2.1 Unique Identifiers

All the classes in the ontology are assigned a unique, seven-digit, zero-padded integer and which is prefixed by “PO:” (Fig. 2a). This identifier (ID) is unique and is not shared by any other ontology, so it is recognizable worldwide. Additionally, the identifiers are never reused, if a term is obsoleted, the ID is retired. If two classes are merged, then one of the IDs becomes an alternate identifier (alt_id). The PO ID corresponds to a universally unique Uniform Resource Locator (URL; http://purl.obolibrary.org/obo/PO_XXXXXXX). These URLs are resolvable via the Ontobee website (<http://www.ontobee.org/index.php>).

2.2.2 Term Names and Synonyms

Classes in the PO are named following the OBO Foundry naming conventions. The preferred term or name (or “label” in OWL format) is designed to be clear and unambiguous, singular, positive, noun form rather than adjective (e.g., “embryo” rather than “embryonic”) and lower case (*see Note 2*). Synonyms are added where needed for several purposes, using one of the standard four scopes: narrow, broad, exact, or related [19] (Fig. 2a).

1. **Narrow:** Used to supply a species-specific names so that users such as plant breeders can relate their terms to the PO hierarchy, e.g., *subterranean tuber axillary vegetative bud* has narrow synonym “potato eye.”
2. **Broad:** Used when the synonym may apply to two or more PO classes, e.g., both *awn* and *trichome* have the broad synonym “bristle.”
3. **Exact:** Used when the same plant structure can have more than one name, e.g., *anther wall* has exact synonym “pollen sac wall.” Additionally, translations into other languages are listed as exact synonyms. The PO lists both Spanish and Japanese translations as exact synonyms; e.g., *anther wall* has exact synonym “pared de la antera” (Spanish) and “葯壁” (Japanese).
4. **Related:** Used when the word or phrase has been used interchangeably with the primary name in the literature, e.g., *phellem* has related synonym “cork.”

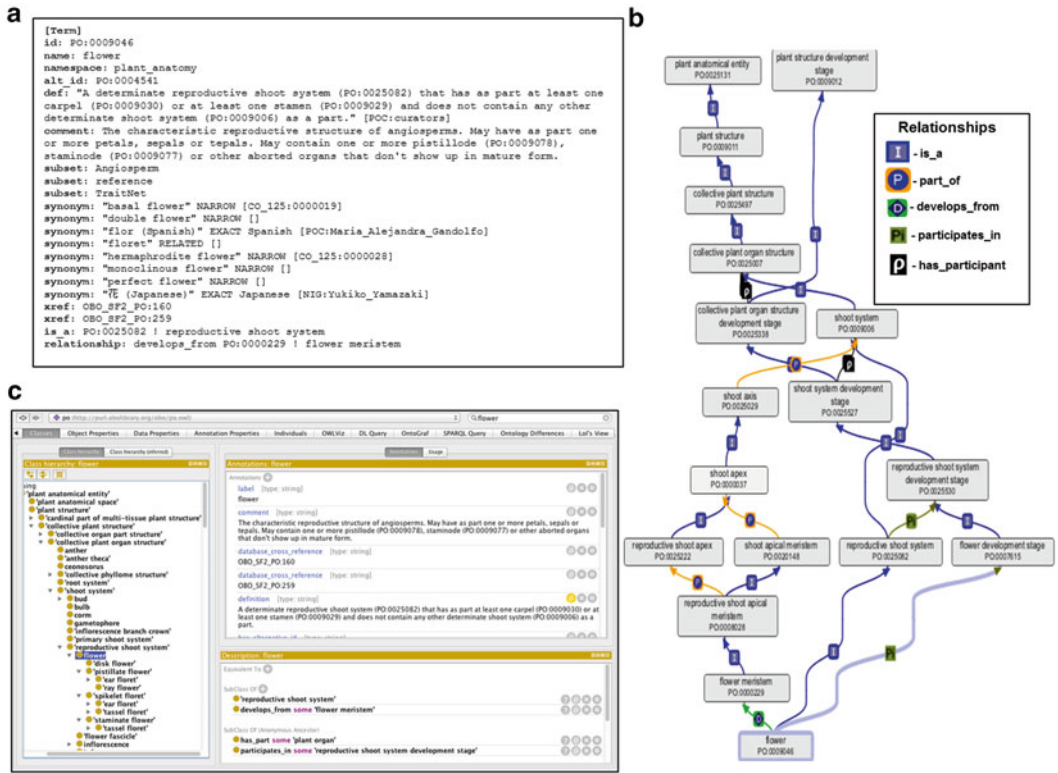


Fig. 2 Elements of an ontology term, for example *flower* PO:0009046. The key elements of a PO term are shown for the PO term *flower*. The identifier (ID) is a unique, seven-digit integer, prefixed by “PO”. The alternate ID is assigned when two terms are merged. Term definitions are carefully written with the assistance of experts in botany and ontology design, and may include references to other ontology terms. The “xref” indicates the related SourceForge tracker, which can be accessed as a link from the term page. (a) The OBO-Edit [49] flat file version of the PO term *flower* shows the key elements in a textual form. (b). The graphical hierarchy for the term *flower*, with the relationships between the two ontology branches shown. Plant structures such as *flower* are linked to the appropriate developmental stage term through the relation *participates_in*. This is read: *flower participates_in flower development stage*, following the direction of the arrow. Images based on the Plant Ontology Version #20 (Released August 2013) and was created using OBO-Edit [49]. (c) Protégé screenshot of the term *flower*. The Protégé software (<http://protege.stanford.edu/>) is used to edit and view the OWL [51] version of the ontology. This view shows the main elements of the PO term *flower*, as well as the class hierarchy on the left hand side

2.2.3 Standardized Definitions

In accordance with the OBO foundry principles, all the classes in the PO have textual (human-readable), English language definitions, while a smaller number also have a “logical” or computer-readable definition.

2.2.4 Definitions in Genus–Differentia Form

The goal is to have all textual definitions structured as Aristotelian definitions [44], which means that they are of the *genus–differentia* form. The *genus* portion comes from the parent class and the

differentia is that characteristic that makes it a unique structure. For example:

flower (PO:0009046)

“A determinate reproductive shoot system (*genus*) that has as part at least one carpel or at least one stamen and does not contain any other determinate shoot system as a part.” (*differentia*)

Many definitions also have one or more comments which may provide additional information or examples (Figs. 2a and 4a). Many of the definitions will also have a literature citation, which may link out to the reference on PubMed or on the PO-Refs page (http://wiki.plantontology.org/index.php/PO_references).

2.2.5 Logical Definitions

The long-term goal of the PO is to have what is known as a “logical definition” [22, 45, 46] for all the classes in the ontology. While a free text definition is useful to human users of the ontology, it is not easily interpreted by a computer. Also known as a “cross product”, a “logical definition” combines two ontology classes in with a specific relation in a *genus-differentia* form, similar to that of the textual definition. Logical definitions are structured in a way that promotes both consistent formulation of the definitions and automatic reasoning provides automated support for reasoning tasks such as classification, debugging, and querying the ontology [47], for example:

embryo plant structure (PO:0025099)

text definition: “A *plant structure* that is part of an *plant embryo*.”

logical definition (cross-product):

intersection_of: PO:0009011 ! *plant structure*

intersection_of: *part_of* PO:0009009 ! *plant embryo*

2.2.6 Relationships Between Ontology Classes

The relationships between the classes are the feature that distinguishes an ontology from a simple glossary (Table 2). These relationships have specifically defined characteristics and domains [48] and along with the classes, they form the basis of a directed acyclic graph [22] (Figs. 1 and 2b). This hierarchical structure is a visualization of the parent-child relationships between the classes at various levels of granularity. The relations allow human users and computers to navigate through the ontology, and query the annotation data. The relations used by the PO are based on the OBO relations ontology (<https://code.google.com/p/obo-relations/>), a modified version (in Web Ontology Language (OWL)) of the Relation Ontology [48]. The most important relations are *is_a* (OWL *subclass_of*) and *part_of*. The PO also currently uses eight other types of relations (Table 2),

Table 2
Relations used in the Plant Ontology (PO)

Relation	Example(s)	Number of assertions
<i>A is_a B</i>	<i>stem is_a shoot axis; epidermis is_a portion of plant tissue</i>	1691
<i>A part_of B</i>	<i>stem internode part_of stem; epidermal cell part_of epidermis</i>	851
<i>A has_part B</i>	<i>inflorescence has_part flower; meristem has_part meristematic cell</i>	56
<i>A derives_by_manipulation_from B</i>	<i>cultured leaf cell derives_by_manipulation_from leaf</i>	6
<i>A develops_from B</i>	<i>apical hook develops_from hypocotyl; trichoblast develops_from epidermal initial</i>	132
<i>A adjacent_to B</i>	<i>anther wall middle layer adjacent_to anther wall endothecium</i>	13
<i>A participates_in B</i>	<i>paraphysis participates_in gametophyte development stage</i>	38
<i>A has_participant B</i>	<i>seed trichome development stage has_participant seed trichome</i>	22
<i>A is located_in B</i>	<i>embryo sac located_in plant ovary ovule</i>	5
<i>A is preceded_by B</i>	<i>floral organ formation stage preceded_by flower meristem transition stage</i>	10

A and *B* represent ontology terms in the PO. The number of assertions (times a relation is used) in the PO is provided in the last column (based on the `plant_ontology_assert.obo` file, Release #20, August 2013: http://www.plantontology.org/docs/release_notes/index.html). For a more detailed description of the relations, see the Relations Wiki page: (http://wiki.plantontology.org/index.php/Relations_in_the_Plant_Ontology)

namely, *has_part*, *derives_by_manipulation_from*, *develops_from*, *adjacent_to*, *participates_in*, *has_participant* and *located_in*, *preceded_by*. A more detailed description can be found on the PO Relations wiki page (http://wiki.plantontology.org/index.php/Relations_in_the_Plant_Ontology).

2.2.7 PO Web Services

The PO offers ontology terms, synonyms, definitions, and comments through two web services (http://plantontology.org/software/po_webservices) to software developers who wish to use the PO in annotation and curation tools for mobile or desktop applications. There are two types of PO web services available at this time: (1) **Term Search**: provides term name and synonym search results, given a partial term name or synonym. (2) **Term Detail**: provides extensive details on multiple pieces of term data, given a PO accession ID.

These services could be used, for example, in applications that allow users to provide PO terms as keywords for image annotation,

gene and phenotype curation, adding markups on scientific literature, and help autofill/autocomplete the database query searches. Future development will include a web service delivering PO annotation data in a similar manner. Full documentation is available on the Plant Ontology website software page: (http://plantontology.org/software/po_webservices) [20].

2.3 Languages and Software Tools for Ontology Development

Historically the PO, like many other ontologies in the OBO Foundry, has been developed in Open Biomedical Ontology (OBO) format (OBOF; http://www.geneontology.org/GO.format.obo-1_2.shtml) using the OBO-Edit software (<http://oboedit.org/>) [49] (Figs. 1 and 2a, b). The OBO format is a human-readable, flat file (Fig. 2a), which models a subset of the OWL format, described above, and can more or less be mapped to OWL [50].

A widely used language for ontology development and representation is the Web Ontology Language (OWL) [22, 51]. OWL is supported by many tools, such as an Application Programmer Interface (OWLAPI) [52] and several tools called reasoners [47], designed to perform automated classification and consistency checking [19]. Protégé is a free, open-source software platform (<http://protege.stanford.edu/>) available for OWL ontology development, with a web version [53], as well as desktop versions (Fig. 2c).

As a convenience to the users of the PO, the ontology files are available in both OBOF and OWL format from the download page: <http://plantontology.org/download>. Future development of the PO will transition to an OWL environment to take advantage of the broader toolset and semantic capabilities. The ontology and the associated annotation data can also be accessed through the ontology web browser (<http://www.plantontology.org/amigo/go.cgi>), which was developed based on that of the Gene Ontology [54].

2.4 Software for Annotation of Image Segments with Ontologies (AISO)

An image segmentation and annotation tool (AISO) has been recently developed in the Jaiswal lab (<http://jaiswallab.cgrb.oregonstate.edu/software/AISO>) for the annotation of images with ontology terms. It is a freely available, Java-based, interactive program that can be run on a curator's desktop and can be used to segment and label digital images. AISO is semantically linked to Plant Ontology terms through a lightweight web service, allowing users and curators to easily select and annotate appropriate plant terms to segmented images. In addition, metadata can be added, such as taxonomic information and collection information. Labeled images can be saved locally and optionally exported to HTML for publication on the web. Future development of AISO will include expansion to include additional ontologies, such as the Plant Trait Ontology (TO) and the Gene Ontology (GO) (see below).

2.5 Other Ontology Resources for Plant Biology

In addition to the PO, there are a number of other ontologies available that may be of interest to plant biologists who wish to annotate plant data. Classes from these cooperating ontologies can be in combination with PO classes for data annotation. Several websites are available that host “look-up” services, such as the European Bioinformatics Institute - Ontology Lookup Service (EBI-OLS) (<http://www.ebi.ac.uk/ontology-lookup/>), National Center for Biomedical Ontology’s (NCBO) Bioportal (<http://bioportal.bioontology.org/ontologies>), and the OBO Foundry (<http://www.obofoundry.org/>).

2.5.1 The Gene Ontology (GO)

As mentioned above, the Gene Ontology (GO; <http://www.geneontology.org/>) is the oldest and most well-known ontology [21, 22]. The domain of the GO covers three very important areas of biological sciences, arranged in three branches; *molecular function*, *cellular component*, and *biological process*. Terms in the *molecular function* branch describe the actions of a gene product at the molecular level. These terms are often used to annotate plant gene products to relate them to their role in the plant. The *cellular-component* branch describes the part of a cell or its extracellular environment in which a gene product is located. The PO and GO cooperate in the description of cellular components, as all plant structures that are part of a *plant cell* are GO classes, for example *chloroplast* (GO:0009507). This prevents any redundancy in the ontologies. The GO *biological process* branch includes classes that describe the collection of molecular events that are related to the functioning of living organisms, and includes many plant-specific classes, for example *seed germination* (GO:0009845). These processes are important markers of plant growth and development and are referenced in many of the definitions in the plant structure development stage (*see Note 1*).

The GO Consortium offers a number of tools and resources to explore and analyze the GO data (http://amigo.geneontology.org/amigo/software_list), including a number of tools to perform term enrichment (<http://www.geneontology.org/page/go-enrichment-analysis>) on a gene list of interest (*see Note 3*).

2.5.2 Plant Trait Ontology (TO)

The Plant Trait Ontology (TO; <http://crop-dev.cgrb.oregonstate.edu/amigo/TO>) was developed as an attempt to move away from the description of phenotypes using free text, which cannot be easily indexed or searched by a computer [35]. The TO describes measurable or observable characteristics of a plant, or part of a plant, or of a stage of plant growth and development. The traits are described by TO classes in a “pre-composed” fashion of an entity and its attribute, and in some cases, a value. The plant entities described are drawn from the PO and plant-relevant portions of the GO, and the attributes and values are terms drawn from the

Phenotypic Qualities Ontology (PATO) (see below). The TO is currently under active development and the future goal of it is for the development of a true reference Plant Trait Ontology [55].

2.5.3 Phenotypic Qualities Ontology (PATO)

The Phenotypic Qualities Ontology (PATO; <http://www.obofoundry.org/cgi-bin/detail.cgi?id=quality>; http://obofoundry.org/wiki/index.php/PATO:Main_Page) provides defined classes to describe a “quality” (*Q*), (which is essentially the combination of an attribute, such as *color*, and a value such as *red*) [56] of an anatomical entity (*E*), such as the PO term *flower*. This method of description is referred to as an “EQ statement” (*phenotype = entity (E) + quality (Q)*) and can be displayed in a human-readable form, where the numeric IDs are not shown. For example, the PO term *flower* can be annotated with the relevant PATO terms; *red* (PATO:0000322) or *yellow* (PATO:0001263), i.e., *E = PO:flower* *Q = PATO:red*.

2.5.4 Ontologies for Describing the Environment

A number of ontologies are being developed to describe environmental conditions, such as the Environment Ontology (EnvO; <http://www.environmentontology.org>) and the Plant Experimental Conditions Ontology (EO; <http://crop-dev.cgrb.oregonstate.edu/amigo/EO>). The EnvO is a controlled, structured vocabulary that is designed to support the annotation of any organism or biological sample with environment descriptors [57]. EnvO contains classes for biomes, environmental features, and environmental material, whereas EO describes the treatments, growth conditions, and/or study types used in various types of plant biology experiments. These classes can be utilized along with the PO to describe the conditions under which a plant was treated in a particular experiment, or its natural environment.

3 Methods

Users can browse the PO to find terms of interest and the associated annotation data, using filters to restrict the results. Alternatively, you can search the ontology directly by term name or by annotation using the search box (Fig. 3a). The Advanced Search (Fig. 3b) feature allows you to search for more than one term or data object type at a time. The SourceForge tracker (Fig. 5) allows users to request new terms or suggest changes to existing terms. The Plant Ontology encourages users to annotate their data with ontology terms and submit their annotations to the PO database. This will increase the visibility of the data and promote sharing among the plant genomics community. Feedback on the ontology and the website is also encouraged.

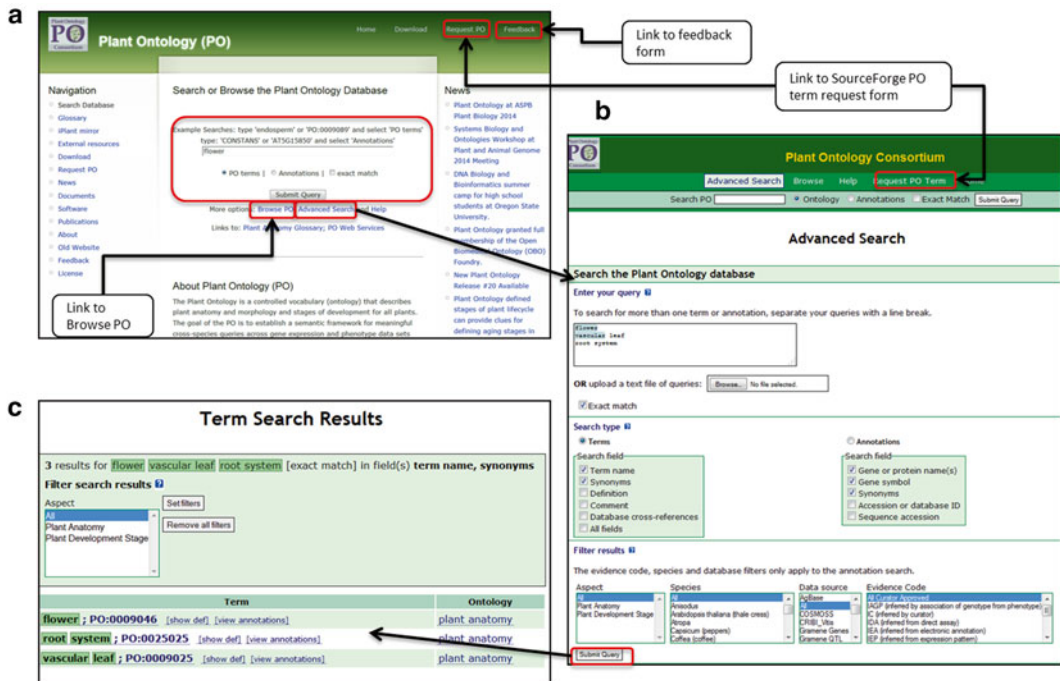


Fig. 3 Browse and search the ontology and annotation database. (a) The Plant Ontology website provides entry points for browsing, and searching and useful links for navigation. Users can enter either term names or annotation data types in the **Search Box** in the center panel. To browse the ontology, the **Browse PO** link will take you to the Tree View pages. The **Navigation menu** on the *left hand side* has links to search the ontology, visit internal and external resources, provide feedback, and request new terms. (b) By selecting **Advanced Search** you can search for more than one term or annotation at a time, or upload a text file with your query terms. (c) On the **Search Results** page, users can select filters and navigate to the page(s) of interest

3.1 Browse the Ontology

1. Start by visiting the PO website: <http://plantontology.org/>
2. To browse the Plant Ontology, select **Browse PO** from the PO home page (Fig. 3a).
3. On the **Term View** page, you can expand nodes marked with a plus sign (+) to see child terms (similar to the **Term Lineage** panel (Fig. 4a)—see below).
4. The type of relationship to the parent term is indicated by the icon to the left of the PO ID. The relationship icons are listed in the legend on the right hand side of the page.
5. You can filter the tree by aspect, i.e., PO branch—either **Plant Anatomy** or **Plant Development Stage**.
6. If you are interested only in annotations from a particular source, you can filter by one or more data source—e.g., TAIR, RAP-DB, MaizeGDB, etc.
7. You can access the list of annotations by clicking on the number in square brackets (see below; Fig. 4a).

4. The **Term Detail** page (Fig. 4a) lists all the information about the term, such as the definition, synonyms, and any comments.
5. The **Term Lineage** panel (Fig. 4a) provides a browsable overview of the ontology hierarchy, with links to the annotation data after the term names—indicated by the number in brackets, e.g., [76,910].
6. Selecting the number in brackets after *flower* will take you to the **Term Annotations** page (Fig. 4b), where you can browse the annotations or filter by species, data source, or evidence code.
7. Selecting the filter **Direct Annotations** restricts the results to those annotated directly to *flower* and ignores any annotations on child terms (*see Note 4*).
8. Links from the annotation objects will take you to additional information such as the **Annotation Information Page**, source references on PubMed, and links out to the source data provider website or database. In some cases, there may be a link out to the GO annotation as well.
9. Selecting some or all of the annotation objects and selecting **Get Annotation Summary** will provide a list of the other PO terms those objects are annotated to.

3.3 Search for a Specific Gene of Interest

1. You can search for a specific gene or an identifier (e.g., *COIN*, *cold inducible*, AT1G05260), or group of genes of interest, for example “*MADS*” genes, by entering the name and selecting **Annotations** below the search box (Fig. 3a).
2. The **Annotation Search Results** page (Fig. 4b) will allow you to filter by species, data source, evidence code, aspect (ontology branch).
3. Links from the individual annotation objects will take you to additional information such as the **Annotation Information Page**, references on PubMed, and links out to the data source (*see Note 5*).

3.4 Advanced Search for a List of Terms or Genes

1. From the PO main page (Fig. 3a) or the top of any page, you can select the **Advanced Search** option (Fig. 3b).
2. From here you can search for more than one term or annotation at a time, or upload a text file with your query terms.
3. Select **Search Type** (Terms or Annotations) and the search fields you wish to include.
4. Select appropriate filters to narrow the results—you can filter by Aspect (branch), species, data source, or evidence code, or a combination of them.
5. The results will be shown on the **Annotation Search Results** (Fig. 4b) page, as described above.

3.5 Request Ontology Terms, or Suggest Changes to Existing Terms Using the SourceForge Term Request Tracker

The PO is a highly collaborative effort, with many groups and individuals who contribute to the development and refinement of the ontology terms and definitions (Table 3). Term requests and comments on existing terms are logged and tracked through the PO SourceForge Term request tracker (<https://sourceforge.net/p/obo/plant-ontology-po-term-requests/>) (Fig. 5). The Tracker provides valuable documentation about when and why changes were made to the PO. There is a link **Request PO** (Fig. 4a) at the top of the any page, or from the Navigation menu on the home page (Fig. 3a), which will take you to Tracker. In addition, the majority of the terms in the PO are associated with one or more trackers, and links from the **External References** panel on the term page (Fig. 4b) will take you to the appropriate tracker page.

Similar SourceForge Term Request Trackers are also used by the other ontologies mentioned here. Term requests and comments for the GO, TO, PATO, EO, and EnvO can be requested by going to the appropriate web page (Table 3). In an effort to prevent spam, you must register on the tracker website and create a login in order to comment or to request a new term, but your login is good for all five of the bio-ontologies listed here (except EnvO; see **Note 6**). This allows users to track their submissions and follow the outcomes, and engages the domain-expert users in the ontology development process.

1. Go to <https://sourceforge.net/p/obo/plant-ontology-po-term-requests/>
2. If you are new to SourceForge, click “Join” to register, or log in (Fig. 5a).
3. To find an existing term request, you can enter a search term in the box. You can restrict the search to a specific group of subjects by clicking the appropriate one from the list on the left hand side, or use a preconfigured search (Fig. 5b).
4. To post a comment on a tracker, type it in the box and click post (Fig. 5c).
5. If you do not find the request you are looking for, you can create a new one using the **Create Ticket** button. Supply a short, self explanatory name such as “New term needed: *Suggested Name*” (Fig. 5d).
6. You can also add a free text label, such as “New term Request.”
7. Click on **Save**
8. The request will be sent as an email to all the PO curators and other interested community members, including the submitter.

Table 3
List of useful links

Name	Link address
Plant Ontology Home page	http://www.plantontology.org/
PO project members	http://plantontology.org/node/220
PO web browser	http://www.plantontology.org/amigo/go.cgi
PO collaborators and contributors	http://plantontology.org/node/8
PO download page	http://plantontology.org/download
PO SourceForge Term tracker	https://sourceforge.net/p/obo/plant-ontology-po-term-requests/
PO Subversion (SVN) repository	http://palea.cgrb.oregonstate.edu/viewsvn/Poc/trunk/associations/
PO Feedback page	http://plantontology.org/contact
PO Relations wiki page	http://wiki.plantontology.org/index.php/Relations_in_the_Plant_Ontology
PO References Wiki page (PO-Refs)	http://wiki.plantontology.org/index.php/PO_references
Details of Release #20, August 2013	http://www.plantontology.org/docs/release_notes/index.html
PO Annotations wiki page	http://plantontology.org/contact http://wiki.plantontology.org/index.php/Category:Annotations
PO web services	http://plantontology.org/software/po_webservices
Annotation of Image Segments with Ontologies (AISO) software	http://jaiswallab.cgrb.oregonstate.edu/software/AISO
IPlant Collaborative	http://www.iplantcollaborative.org/
IPlant mirror site of Plant Ontology	http://iplant.plantontology.org/
Open Biological and Biomedical Ontologies (OBO) Foundry	http://obofoundry.org
OBO Foundry Initiative	http://www.obofoundry.org/about.shtml
OBO Foundry Reviews	http://www.obofoundry.org/wiki/index.php/Results_of_OBO_Foundry_Reviews
Gene Ontology (GO)	http://www.geneontology.org/
GO SourceForge Term tracker	http://sourceforge.net/p/geneontology/ontology-requests/
Go Consortium tools and resources	http://amigo.geneontology.org/amigo/software_list
GO Term Enrichment Tool	http://www.geneontology.org/page/go-enrichment-analysis
Gene Ontology annotation file format (gaf 2 format)	http://www.geneontology.org/page/go-annotation-file-gaf-format-20
Plant Trait Ontology (TO) Browser	http://crop-dev.cgrb.oregonstate.edu/amigo/TO
TO SourceForge Term Tracker	http://sourceforge.net/p/obo/plant-trait-ontology-to-requests/

(continued)

Table 3
(continued)

Name	Link address
The Plant Experimental Conditions Ontology (EO)	http://crop-dev.cgrb.oregonstate.edu/amigo/EO
EO SourceForge Term Tracker	http://sourceforge.net/p/obo/plant-environment-ontology-eo/
Phenotypic Qualities Ontology (PATO)	http://www.obofoundry.org/cgi-bin/detail.cgi?id=quality
PATO Wiki Main Page	http://obofoundry.org/wiki/index.php/PATO:Main_Page
PATO SourceForge Term tracker	http://sourceforge.net/p/obo/phenotypic-quality-pato-requests/
Environment Ontology (EnvO)	http://www.environmentontology.org
EnvO Google Code Issues Tracker	https://code.google.com/p/envo/issues/list
European Bioinformatics Institute - Ontology Lookup Service (EBI-OLS)	http://www.ebi.ac.uk/ontology-lookup/
National Center for Biomedical Ontology (NCBO) Bioportal	http://bioportal.bioontology.org/ontologies
Ontobee	http://www.ontobee.org/index.php
OBO relations ontology	https://code.google.com/p/obo-relations/
Protégé website	http://protege.stanford.edu/
OBO-Edit software	http://oboedit.org/
OBO Flat File Format Specification, version 1.2	http://www.geneontology.org/GO.format.obo-1_2.shtml
The Arabidopsis Information Resource (TAIR)	https://www.arabidopsis.org/
MaizeGenetics and GenomicsDatabase (MaizeGDB)	http://www.maizegdb.org/
<i>Physcomitrella patens</i> Resource (cosmoss)	http://www.cosmoss.org/
Solanaceae Genome Network (SGN)	http://solgenomics.net/
Gramene: A comparative resource for plants	http://www.gramene.org/
Nottingham Arabidopsis Stock Centre (NASC)	http://arabidopsis.info/
MaizeGenetics Cooperation Stock Center (MGCSC)	http://maizecoop.cropsci.uiuc.edu/
AgBase	http://www.agbase.msstate.edu/

(continued)

Table 3
(continued)

Name	Link address
Grape GenomeDatabase (hosted at CRIBI)	http://genomes.cribi.unipd.it/grape/
The Rice Annotation Project (RAP-DB)	http://rapdb.dna.affrc.go.jp/
Ensembl Plants	http://plants.ensembl.org
Phytozome	http://phytozome.jgi.doe.gov/pz/portal.html#
Gigascience	http://www.gigasciencejournal.com/
Sequence Read Archive (SRA)	http://www.ncbi.nlm.nih.gov/sra
Transcriptome Shotgun Assembly Sequence Database (TSA)	http://www.ncbi.nlm.nih.gov/genbank/tsa
Gene Expression Omnibus (GEO)	http://www.ncbi.nlm.nih.gov/geo/
ArrayExpress	http://www.ebi.ac.uk/arrayexpress/
Dryad	http://datadryad.org/

Links to various pages and resources mentioned in the text, including collaborating plant database groups that contribute substantial annotation data to the PO database

3.6 Contribute Plant Genomic Data Annotations to the Plant Ontology Database

1. Select appropriate data of interest—the best is gene expression, or mutant gene studies where careful attention has been paid to the selection of plant samples and stages of development.
2. Contact the Plant Ontology curators (see below) to collaborate on efforts to annotate the relevant study. They can answer questions and help select the appropriate plant structure and developmental stage terms.
3. Create a mapping table between the terms used in the study and the relevant PO terms (see examples on <http://wiki.plantontology.org/index.php/Category:Annotations>). Determine PO annotations for the *plant structure developmental stages*, as well as the *whole plant developmental stages*.
4. Request any needed PO terms and fix any issues with existing terms and definitions for your specific plant species, by using the PO SourceForge Tracker (see above).
5. In collaboration with the PO curation team, help create appropriate database cross-references (dbxrefs) to the expression data—e.g., MaizeGDB, TAIR, CRIBI, RAPDB, or other source, so that the PO can provide a link to the data. If you do not have an online database, the PO curation team can assist you by suggesting archiving sites.

a Register or Login to SourceForge

Log in or register on SourceForge site

b Search for existing request(s)

Enter search term

Filter by group

Preset searches

c Comment on an existing request

Enter comment in box and click on "Post"

d. Create a new term request tracker

Select "Create Ticket" and enter a short name

Enter proposed definition and any other information, or comment

Fig. 5 The SourceForge PO term request tracker. The PO SourceForge PO term request tracker (<https://sourceforge.net/p/obo/plant-ontology-po-term-requests/>) is the main site for requesting new terms or suggesting changes. **(a)** If you are new to SourceForge, click "Join" to register, or else log in. **(b)** To find an existing tracker, you can enter a search term in the box. You can restrict the search to a specific group of subjects by clicking the appropriate one from the list on the *left hand side*, or use a preconfigured search. **(c)** To post a comment on a tracker, type it in the box and click "post". **(d)** If you do not find the ticket you are looking for, you can create a new one using the **Create Ticket** button. Supply a short, self explanatory name such as "New term needed: *Suggested Name*". You can also add a free text label, such as "New term Request"

6. Create the appropriate 16 column spreadsheet listing the PO IDs and the associated genes, gene models, proteins, etc., using the Gene Ontology annotation file format (gaf 2 format; <http://www.geneontology.org/page/go-annotation-file-gaf-format-20>). Usually, we recommend authors make one file for anatomy terms, and one for development terms.
7. Use column 16 to add additional information to the annotation, if needed. These can be notations about plant structures that are *part_of* another plant structure, or that occur during a particular growth stage. For example, a gene expressed in a *leaf base* should also have a column 16 annotation to the specific type of leaf (e.g., *leaf base part_of vascular leaf*). These should be added as an additional line in the annotation file, directly on

the specific parent term, as the annotations will generally not propagate through these on the browser.

8. Transfer the files to tab delimited text file and upload to the PO Subversion (SVN) repository (<http://palea.cgrb.oregonstate.edu/viewsvn/Poc/trunk/associations/>). If you are a regular contributor, the PO curation team will help you set up access to the PO SVN repository. Otherwise, the annotation files can be sent to the PO curator who is handling the submission.
9. Annotations will be tested on the PO-beta site with the appropriate ontology version to test the links and the file loading.
10. If all is well, the new annotations will be loaded with the rest of the annotation data at the next PO release, and will be accessible through the PO website.

3.7 Providing Feedback

1. Start by visiting the PO website: <http://plantontology.org/>.
2. Click on the **Feedback** link (<http://plantontology.org/contact>) at the top of the home page (Fig. 3a) or in the **Navigation Menu**.
3. Fill in the required information (name, email address, subject) and select the appropriate category (feedback on ontology or website).
4. You can provide additional details in the **Message** box.
5. Fill in the code depicted in the image. This is required to prevent spam.
6. Click on the **Send e-mail** button.
7. The message will be sent to the PO curation team, and you can expect a response within 2–3 business days.

4 Notes

1. The *plant structure developmental stages* does not define the developmental processes which occur during the stages, as these are described in the *biological process* branch of the Gene Ontology.
2. The PO works to resolve issues where the same name is used to describe different plant structures. For example, the term *leaf* is commonly used to describe the *vascular leaf* structure found in angiosperms, gymnosperms, and ferns, as well as the similar leaf-like non-vascular structure called a *phyllid* found in bryophytes. In order to differentiate the vascular and non-vascular types of leaf structures, we created the general parent term *leaf* and created two child terms, *non-vascular leaf* (synonym: phyllid) and *vascular leaf*. The term *non-vascular leaf* has no *is_a* child terms in the PO.

3. Unlike the GO enrichment tool offered by the Gene Ontology for finding and/or summarizing genome annotation and gene expression data, the PO does not provide such a tool, but we are looking at ways to implement it in the future.
4. When you search for annotations on a particular PO term, the browser returns annotations on any *is_a* and *part_of* child terms as well. They are listed separately in the browser window.
5. Similar to many other websites, on the PO website many of the pieces of information are hyperlinked to additional pages, and in most cases, there are more than one way to get to the various pages of results.
6. The EnvO tracker is not on SourceForge, but on a Google Code (listed in Table 3). You can use a Gmail or Google account as the login for this site.

Acknowledgements

The authors would like to acknowledge the contributions of the current and former members of the PO Project (<http://plantontology.org/node/220>) for their contributions to ontology development; the Gene Ontology Consortium (www.geneontology.org) for its leadership in the ontology field, for sharing software tools AmiGO an ontology browser, and the GO database package, which were both customized for the PO project; Christopher Sullivan (Center for Genome Research and Biocomputing at Oregon State University) for help with hosting and maintenance of the PO project web servers, and the members of the Jaiswal lab group at Oregon State University. We also acknowledge the iPlant Collaborative (<http://www.iplantcollaborative.org/>) for hosting a mirror site of PO (<http://iplant.plantontology.org/>). We extend special thanks to the numerous collaborators and domain experts (<http://plantontology.org/node/8>) who continue to contribute to the development and maintenance of the PO and the annotation database. This work was supported by the US National Science Foundation (Award # [IOS:0822201 award](#)).

References

1. Michael TP, Jackson S (2013) The first 50 plant genomes. *Plant Genome* 6:1–7. doi:[10.3835/plantgenome2013.03.0001in](https://doi.org/10.3835/plantgenome2013.03.0001in)
2. Fox SE, Geniza M, Hanumappa M et al (2014) De novo transcriptome assembly and analyses of gene expression during photomorphogenesis in diploid wheat *Triticum monococcum*. *PLoS One* 9:e96855. doi:[10.1371/journal.pone.0096855](https://doi.org/10.1371/journal.pone.0096855)
3. Fox SE, Preece J, Kimbrel JA et al (2013) Sequencing and de novo transcriptome assembly of *Brachypodium sylvaticum* (Poaceae). *Appl Plant Sci* 1:1200011. doi:[10.3732/apps.1200011](https://doi.org/10.3732/apps.1200011)

4. Mishima K, Fujiwara T, Iki T et al (2014) Transcriptome sequencing and profiling of expressed genes in cambial zone and differentiating xylem of Japanese cedar (*Cryptomeria japonica*). *BMC Genomics* 15:219
5. Mudalkar S, Golla R, Ghatty S, Reddy A (2014) De novo transcriptome analysis of an imminent biofuel crop, *Camelina sativa* L. using Illumina GAIIX sequencing platform and identification of SSR markers. *Plant Mol Biol* 84:159–171. doi:[10.1007/s11103-013-0125-1](https://doi.org/10.1007/s11103-013-0125-1)
6. Ranjan A, Ichihashi Y, Farhi M et al (2014) De novo assembly and characterization of the transcriptome of the parasitic weed *Cuscuta pentagona* identifies genes associated with plant parasitism. *Plant Physiol* 166:1186. doi:[10.1104/pp.113.234864](https://doi.org/10.1104/pp.113.234864)
7. Wu J, Xu Z, Zhang Y et al (2014) An integrative analysis of the transcriptome and proteome of the pulp of a spontaneous late-ripening sweet orange mutant and its wild type improves our understanding of fruit ripening in citrus. *J Exp Bot* 65:1651–1671. doi:[10.1093/jxb/cru044](https://doi.org/10.1093/jxb/cru044)
8. The Arabidopsis Genome Initiative (2000) Analysis of the genome sequence of the flowering plant *Arabidopsis thaliana*. *Nature* 408:796–815. doi:[10.1038/35048692](https://doi.org/10.1038/35048692)
9. Myburg AA, Grattapaglia D, Tuskan GA et al (2014) The genome of *Eucalyptus grandis*. *Nature* 510:356. doi:[10.1038/nature13308](https://doi.org/10.1038/nature13308)
10. Nystedt B, Street NR, Wetterbom A et al (2013) The Norway spruce genome sequence and conifer genome evolution. *Nature* 497: 579–584
11. Schmutz J, McClean PE, Mamidi S et al (2014) A reference genome for common bean and genome-wide analysis of dual domestications. *Nat Genet* 46:707
12. Sansone S-A, Rocca-Serra P, Field D et al (2012) Toward interoperable bioscience data. *Nat Genet* 44:121–126. doi:[10.1038/ng.1054](https://doi.org/10.1038/ng.1054)
13. Kolker E, Özdemir V, Martens L et al (2014) Toward more transparent and reproducible omics studies through a common metadata checklist and data publications. *OMICS J Integr Biol* 18:10–14. doi:[10.1089/omi.2013.0149](https://doi.org/10.1089/omi.2013.0149)
14. Wruck W, Peuker M, Regenbrecht CRA (2014) Data management strategies for multinational large-scale systems biology projects. *Brief Bioinform* 15:65–78. doi:[10.1093/bib/bbs064](https://doi.org/10.1093/bib/bbs064)
15. Taylor CF, Field D, Sansone S-A et al (2008) Promoting coherent minimum reporting guidelines for biological and biomedical investigations: the MIBBI project. *Nat Biotechnol* 26:889–896. doi:[10.1038/nbt0808-889](https://doi.org/10.1038/nbt0808-889)
16. Rocca-Serra P, Brandizi M, Maguire E et al (2010) ISA software suite: supporting standards-compliant experimental annotation and enabling curation at the community level. *Bioinformatics* 26:2354–2356. doi:[10.1093/bioinformatics/btq415](https://doi.org/10.1093/bioinformatics/btq415)
17. Appels R, Nystrom-Persson J, Keeble-Gagnere G (2014) Advances in genome studies in plants and animals - Springer. *Funct Integr Genomics* 14:1–9. doi:[10.1007/s10142-014-0364-5](https://doi.org/10.1007/s10142-014-0364-5)
18. Stevens R, Rector A, Hull D (2010) What is an ontology? *Ontogenesis*
19. Walls RL, Athreya B, Cooper L et al (2012) Ontologies as integrative tools for plant science. *Am J Bot* 99:1263–1275. doi:[10.3732/ajb.1200222](https://doi.org/10.3732/ajb.1200222)
20. Cooper L, Walls RL, Elser J et al (2013) The Plant Ontology as a tool for comparative plant anatomy and genomic analyses. *Plant Cell Physiol* 54:e1. doi:[10.1093/pcp/pcs163](https://doi.org/10.1093/pcp/pcs163)
21. Ashburner M, Ball CA, Blake JA et al (2000) Gene ontology: tool for the unification of biology. *Nat Genet* 25:25–29. doi:[10.1038/75556](https://doi.org/10.1038/75556)
22. The Gene Ontology Consortium (2013) Gene ontology annotations and resources. *Nucleic Acids Res* 41:D530–D535. doi:[10.1093/nar/gks1050](https://doi.org/10.1093/nar/gks1050)
23. Schaeffer ML, Harper LC, Gardiner JM et al (2011) MaizeGDB: curation and outreach go hand-in-hand. *Database (Oxford)* 2011: bar022. doi:[10.1093/database/bar022](https://doi.org/10.1093/database/bar022)
24. Lamesch P, Berardini TZ, Li D et al (2012) The Arabidopsis Information Resource (TAIR): improved gene annotation and new tools. *Nucleic Acids Res* 40:D1202–D1210. doi:[10.1093/nar/gkr1090](https://doi.org/10.1093/nar/gkr1090)
25. Monaco MK, Stein J, Naithani S et al (2014) Gramene 2013: comparative plant genomics resources. *Nucleic Acids Res* 42:D1193–D1199. doi:[10.1093/nar/gkt1110](https://doi.org/10.1093/nar/gkt1110)
26. Kersey PJ, Allen JE, Christensen M et al (2014) Ensembl Genomes 2013: scaling up access to genome-wide data. *Nucleic Acids Res* 42:D546–D552. doi:[10.1093/nar/gkt979](https://doi.org/10.1093/nar/gkt979)
27. Neale DB, Wegrzyn JL, Stevens KA et al (2014) Decoding the massive genome of loblolly pine using haploid DNA and novel assembly strategies. *Genome Biol* 15:R59. doi:[10.1186/gb-2014-15-3-r59](https://doi.org/10.1186/gb-2014-15-3-r59)
28. Shulaev V, Sargent DJ, Crowhurst RN et al (2011) The genome of woodland strawberry (*Fragaria vesca*). *Nat Genet* 43:109–116. doi:[10.1038/ng.740](https://doi.org/10.1038/ng.740)

29. Varshney RK, Mir RR, Bhatia S et al (2014) Integrated physical, genetic and genome map of chickpea (*Cicer arietinum* L.). *Funct Integr Genomics* 14:59–73. doi:[10.1007/s10142-014-0363-6](https://doi.org/10.1007/s10142-014-0363-6)
30. Rowley ER, Fox SE, Bryant DW et al (2012) Assembly and characterization of the European Hazelnut “Jefferson” transcriptome. *Crop Sci* 52:2679. doi:[10.2135/cropsci2012.02.0065](https://doi.org/10.2135/cropsci2012.02.0065)
31. Sharma N, Jung C-H, Bhalla PL, Singh MB (2014) RNA sequencing analysis of the gametophyte transcriptome from the liverwort, *marchantia polymorpha*. *PLoS One* 9:e97497
32. Liu M, Qiao G, Jiang J et al (2012) Transcriptome sequencing and de novo analysis for ma bamboo (*Dendrocalamus latiflorus* Munro) using the Illumina platform. *PLoS One* 7:e46766. doi:[10.1371/journal.pone.0046766](https://doi.org/10.1371/journal.pone.0046766)
33. Garcia-Hernandez M, Berardini TZ, Chen G et al (2002) TAIR: a resource for integrated *Arabidopsis* data. *Funct Integr Genomics* 2:239–253. doi:[10.1007/s10142-002-0077-z](https://doi.org/10.1007/s10142-002-0077-z)
34. Vincent L, Coe EH, Polacco ML (2003) *Zea mays* ontology - a database of international terms. *Trends Plant Sci* 8:517–520. doi:[10.1016/j.tplants.2003.09.014](https://doi.org/10.1016/j.tplants.2003.09.014)
35. Jaiswal P, Ware D, Ni J et al (2002) Gramene: development and integration of trait and gene ontologies for rice. *Comp Funct Genom* 3:132–136
36. Jaiswal P, Avraham S, Ilic K et al (2005) Plant Ontology (PO): a controlled vocabulary of plant structures and growth stages. *Comp Funct Genom* 6:388–397
37. Ilic K, Kellogg EA, Jaiswal P et al (2007) The plant structure ontology, a unified vocabulary of anatomy and morphology of a flowering plant. *Plant Physiol* 143:587–599. doi:[10.1104/pp.106.092825](https://doi.org/10.1104/pp.106.092825)
38. Pujar A, Jaiswal P, Kellogg EA et al (2006) Whole-plant growth stage ontology for Angiosperms and its application in plant biology. *Plant Physiol* 142:414–428. doi:[10.1104/pp.106.085720](https://doi.org/10.1104/pp.106.085720)
39. Rosse C, Mejino JLV (2007) The foundational model of anatomy ontology. In: Burger A, Davidson D, Baldock R (eds) *Anatomy ontologies for bioinformatics: principles and practice*. Springer, New York, NY, pp 59–117
40. Haendel M, Neuhaus F, Osumi-Sutherland D et al (2008) CARO - the common anatomy reference ontology. In: Burger A, Davidson D, Baldock R (eds) *Anatomy ontologies for bioinformatics: principles and practice*. Springer, New York, NY, pp 327–349
41. O’Donoghue M-T, Chater C, Wallace S et al (2013) Genome-wide transcriptomic analysis of the sporophyte of the moss *Physcomitrella patens*. *J Exp Bot* 64:3567–3581. doi:[10.1093/jxb/ert190](https://doi.org/10.1093/jxb/ert190)
42. Hill DP, Smith B, McAndrews-Hill MS, Blake J (2008) Gene Ontology annotations: what they mean and where they come from. *BMC Bioinform* 9:S2
43. Smith B, Ashburner M, Rosse C et al (2007) The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nat Biotechnol* 25:1251–1255. doi:[10.1038/nbt1346](https://doi.org/10.1038/nbt1346)
44. Rosse C, Mejino JLV Jr (2003) A reference ontology for biomedical informatics: the foundational model of anatomy. *J Biomed Inform* 36:478–500. doi:[10.1016/j.jbi.2003.11.007](https://doi.org/10.1016/j.jbi.2003.11.007)
45. Meehan T, Masci A, Abdulla A et al (2011) Logical development of the cell ontology. *BMC Bioinform* 12:6
46. Mungall CJ, Bada M, Berardini TZ et al (2011) Cross-product extensions of the gene ontology. *J Biomed Inform* 44:80–86. doi:[10.1016/j.jbi.2010.02.002](https://doi.org/10.1016/j.jbi.2010.02.002)
47. Dentler K, Cornet R, ten Teije A, de Keizer N (2011) Comparison of reasoners for large ontologies in the OWL 2 EL profile. *Semant Web* 2:71–87. doi:[10.3233/SW-2011-0034](https://doi.org/10.3233/SW-2011-0034)
48. Smith B, Ceusters W, Klagges B et al (2005) Relations in biomedical ontologies. *Genome Biol* 6:R46
49. Day-Richter J, Harris MA, Haendel M et al (2007) OBO-Edit an ontology editor for biologists. *Bioinformatics* 23:2198–2200. doi:[10.1093/bioinformatics/btm112](https://doi.org/10.1093/bioinformatics/btm112)
50. Tirmizi S, Aitken S, Moreira D et al (2011) Mapping between the OBO and OWL ontology languages. *J Biomed Semant* 2:S3
51. Horridge M, Drummond N, Goodwin J, et al. (2006) The Manchester OWL Syntax. *Proc. 2006 OWL Exp. Dir. Workshop OWL-ED2006*
52. Horridge M, Bechhofer S (2011) The OWL API: a Java API for OWL ontologies. *Semant Web* 2:11–21. doi:[10.3233/SW-2011-0025](https://doi.org/10.3233/SW-2011-0025)
53. Horridge M, Tudorache T, Nuytas C et al (2014) WebProtégé: a collaborative web based platform for editing biomedical ontologies. *Bioinformatics* 30:2384. doi:[10.1093/bioinformatics/btu256](https://doi.org/10.1093/bioinformatics/btu256)
54. Carbon S, Ireland A, Mungall CJ et al (2009) AmiGO: online access to ontology and annotation data. *Bioinformatics* 25:288–289. doi:[10.1093/bioinformatics/btn615](https://doi.org/10.1093/bioinformatics/btn615)

55. Arnaud E, Cooper L, Shrestha R, et al. (2012) Towards a reference Plant Trait Ontology for modeling knowledge of plant traits and phenotypes. Proceedings of the International Conference on Knowledge Engineering and Ontology Development, Barcelona, Spain, pp 220–225
56. Gkoutos G, Green E, Mallon A-M et al (2004) Using ontologies to describe mouse phenotypes. *Genome Biol* 6:R8
57. Buttigieg PL, Morrison N, Smith B et al (2013) The environment ontology: contextualising biological and biomedical entities. *J Biomed Semant* 4:43

Chapter 6

Ensembl Plants: Integrating Tools for Visualizing, Mining, and Analyzing Plant Genomics Data

Dan Bolser, Daniel M. Staines, Emily Pritchard, and Paul Kersey

Abstract

Ensembl Plants (<http://plants.ensembl.org>) is an integrative resource presenting genome-scale information for a growing number of sequenced plant species (currently 33). Data provided includes genome sequence, gene models, functional annotation, and polymorphic loci. Various additional information are provided for variation data, including population structure, individual genotypes, linkage, and phenotype data. In each release, comparative analyses are performed on whole genome and protein sequences, and genome alignments and gene trees are made available that show the implied evolutionary history of each gene family. Access to the data is provided through a genome browser incorporating many specialist interfaces for different data types, and through a variety of additional methods for programmatic access and data mining. These access routes are consistent with those offered through the Ensembl interface for the genomes of non-plant species, including those of plant pathogens, pests, and pollinators.

Ensembl Plants is updated 4–5 times a year and is developed in collaboration with our international partners in the Gramene (<http://www.gramene.org>) and transPLANT projects (<http://www.transplantdb.org>).

Key words Databases, Genome browser, Genomics, Transcriptomics, Functional genomics, Comparative genomics, Genetic variation, Phenotype, Crops, Cereals

1 Introduction

Against a backdrop of probable population growth and environmental degradation, humankind needs to improve the efficiency and sustainability of land use. A potential avenue to achieve this is crop improvement, where significant advances may be possible if we can exploit new knowledge of genetic potential obtained through the use of large-scale technologies for nucleotide sequencing and phenotyping. Genome-wide association studies (GWAS) can translate the raw data from these approaches into molecular quantitative trait loci (QTLs) and variant based markers, which can be used to facilitate crop improvement via methods such as marker assisted breeding [1], genomic selection [2], association genetics [3], genetic modification [4] and, where appropriate, genome editing [5].

Driven by this need, and facilitated by ongoing improvements in the available technology, the number of newly sequenced plant genomes is growing exponentially year on year, with approximately 30 new plant genomes sequenced in 2012 [6]. However, a relatively small number of crop species together account for a very large fraction of global agronomic output. For example, 50 % of global crop production in tonnes can be accounted for by just four crops: wheat, rice, maize, and sugar cane [7]. The top 20 cultivated crop species comprise more than 80 % of production, 6.6 out of 8 billion tonnes produced globally in 2011. It is likely, therefore, that the genomes of all economically important crops will be sequenced, assembled, and annotated in the near future.

Ensembl Plants is a part of Ensembl Genomes [8], a genome centric resource, that utilizes reference genome sequences as a framework to integrate variant, functional, expression, marker, and comparative data for a number of plant species through a consistent set of interactive and programmatic interfaces. The organization of the data and the provision of an associated toolset aim to facilitate basic and translational biological research. In particular, the development of a reference catalogue for plant genetic diversity, and its linkage to phenotypic data, is badly needed to accelerate crop improvement programs and an increasing range of data sets have been incorporated into Ensembl Plants for this purpose.

Ensembl Plants is developed in collaboration with the Gramene project (<http://www.gramene.org>) [9] in the United States and we collaborate closely with 10 important European genomics and informatics groups in the transPLANT project (<http://www.transplantdb.org>). Through these efforts, we aim to use common, reference, traceable data across multiple portals with different remits and user communities.

2 Materials

2.1 Database Schema and Structure

Ensembl Plants is primarily implemented in the open-source relational database management system MySQL, a data storage engine designed to support data consistency and flexible views. The overall data structure is modular, with different data (e.g., core annotation, comparative genomics, functional genomics, variation data) modeled by distinct schemas. The complete release comprises a separate database instance for each reference genome for each module for which the relevant data type is available.

The core genomics schema is modeled on the central dogma of biology, linking genome sequence to genes, transcripts, and their translations, each of which can be decorated with functional annotation. Much annotation in Ensembl Plants takes the form of cross references, reciprocal web links to entries in other resources that either represent the primary source of the biological entity shown

in Ensembl or which may provide additional information about it. Cross references to entries in external resources describing functional entities such as domains, reactions, and processes allow these entries to serve as controlled vocabularies for functional annotation within Ensembl. Ancillary tables in the core schema keep track of identifiers between successive versions of the genome assembly and gene build. The schemas for specialist data types each contain a copy of the most important tables in the core schema (which allows efficient querying), together with additional, domain-specific tables. This model allows for rapid schema evolution where necessary, for example in data domains where the available information is itself in a state of rapid flux, without decreasing the stability of the core schema.

The databases can be downloaded for local installation or alternatively a public MySQL server provides access. Programmatic access is supported through the existence of two Application Programming Interfaces, which allow users to discover and access data through an abstraction layer that hides the detailed structure of the underlying data store. One is written for the Perl programming language, while the other uses the language-agnostic Representational State Transfer (REST) paradigm.

Interactive access is provided through a multifunctional genome browser. In addition to displaying data from the associated schemas, the browser can also be configured to access external data files, which can improve response times when querying large data, and which additionally allow users to visualize their own data in the context of the public reference. A list of data formats and types that can be uploaded to the browser is given in Table 1.

In addition to the primary databases, Ensembl Plants also provides access to denormalized data warehouses, constructed using the BioMartBioMart tool kit [10]. These are specialized databases optimized to support the efficient performance of common gene- and variant-centric queries, and can be accessed through their own web-based and programmatic interfaces. Finally, a variety of data selections are exported from the databases in common file formats and made available for user download via the file transfer protocol (FTP).

2.2 Overview of Data Content

2.2.1 Reference Genomes and Associated Data

The set of genomes currently included in Ensembl Plants is given in Table 2. Generally, gene model annotations are imported from the relevant authority for each species (*see* references in Table 2). After import, various automatic computational analysis are performed for each genome. A summary of these is given in Table 3. Additionally, specific datasets are imported and analyzed according to the requirements of individual user communities. These datasets typically fall into two classes, sequence alignments and derived positional features, such as variant loci. Variation datasets incorporated are listed in Table 4. Details of other datasets incorporated can be found through the homepage for each species within the Ensembl Plants portal.

Table 1

List of formats currently supported for user-supplied data. For details, see <http://plants.ensembl.org/info/website/upload/index.html>

Format	Type of data (and notes)
BAM	Sequence alignments (no upload required, index required). http://plants.ensembl.org/info/website/upload/large.html#bam-format
BED	Genes and features. http://plants.ensembl.org/info/website/upload/bed.html
BedGraph	Continuous-valued data. http://plants.ensembl.org/info/website/upload/bed.html#bedGraph
BigBed	Genes and features (no upload required, indexed BED). http://plants.ensembl.org/info/website/upload/large.html#bb-format
BigWig	Continuous-valued data (no upload required). http://plants.ensembl.org/info/website/upload/large.html#bw-format
DAS	Genes and features. http://plants.ensembl.org/Help/Glossary?id=77
Generic	Genes and features. http://plants.ensembl.org/info/website/upload/generic.html
GFF/GTF	Genes and features. http://plants.ensembl.org/info/website/upload/gff.html
PSL	Sequence alignments. http://plants.ensembl.org/info/website/upload/psl.html
TrackHub	Collections of tracks. http://plants.ensembl.org/info/website/upload/large.html#hubs
VEP	Variation coding consequences. http://plants.ensembl.org/info/website/upload/var.html
VCF	Variants (no upload required, index required). http://plants.ensembl.org/info/website/upload/large.html#vcf-format
WIG	Continuous-valued data. http://plants.ensembl.org/info/website/upload/wig.html

2.2.2 Core Functional Annotation

The program InterProScan [11] is used to predict the domain structure for each predicted protein sequence. In addition, genes are annotated with functional information using terms from the Gene Ontology, Plant Ontology, and other relevant ontologies. Names and descriptions imported from the most authoritative source for each genome and cross references to relevant objects in other databases are added.

2.2.3 Variation

The Ensembl Plants variation module is able to store: variant loci and their known alleles, including single nucleotide polymorphisms, indels, and structural variations; the functional consequence of known variants on protein-coding genes; and individual

Table 2
List of genomes currently available in Ensembl Plants

Species	Brief description	Chr/Pan	Size Mb	No. genes
<i>Amborella trichopoda</i>	An important evolutionary reference point in the evolution of plants [22]	No P	706	27,313
<i>Arabidopsis lyrata</i>	A close relative of <i>A. thaliana</i> making a useful evolutionary reference [23]	Yes	207	32,667
<i>Arabidopsis thaliana</i>	A model plant [24]	Yes P	120	27,416
<i>Musa acuminata</i>	Banana is an economically important food crop and the first non-grass monocot genome to be sequenced, providing an important data point for evolutionary comparison [25]	Yes	473	36,525
<i>Hordeum vulgare</i>	Barley is an economically important crop and an important model of environmental diversity for development of wheat [26]	Yes	4706	24,211
<i>Brachypodium distachyon</i>	A model cereal [27]	Yes	272	26,552
<i>Brassica rapa</i>	Representative of an economically important and evolutionarily interesting group of vegetable crops [28]	Yes	284	41,018
<i>Chlamydomonas reinhardtii</i>	A model green algal genome and evolutionary reference point in the evolution of plants [29]	No P	120	14,416
<i>Cyanidioschyzon merolae</i>	A model red algal genome and evolutionary reference point in the evolution of plants [30]	Yes P	16	5009
Grape <i>Vitis vinifera</i>	An economically important crop and model dicot genome [31]	Yes P	486	29,971
Maize <i>Zea mays</i>	An economically important crop, accounting for over 10 % of global agricultural production [32]	Yes	2067	39,475
<i>Medicago truncatula</i>	A model organism for legume biology [33]	Yes	314	44,115
<i>Setaria italica</i>	Millet is an economically important food crop and model of C4 photosynthesis [34]	Yes	406	35,471
<i>Physcomitrella patens</i>	A model moss genome and evolutionary reference point in the evolution of plants [35]	No P	480	32,273

(continued)

Table 2
(continued)

Species	Brief description	Chr/Pan	Size Mb	No. genes
<i>Populus trichocarpa</i>	Poplar is an economically important source of timber and a model tree [36]	Yes	417	41,377
<i>Solanum tuberosum</i>	Potato is an economically important food crop, accounting for approximately 5 % of global agricultural production [37]	Yes	811	39,021
<i>Prunus persica</i>	Peach is an economically important deciduous fruit tree in the Rosaceae family [38]	No	227	28,087
Rices	An economically important food crop, accounting for nearly 10 % of global agricultural production			
<i>Oryza brachyantha</i>	A disease resistant wild rice [39]	Yes	261	32,038
<i>Oryza glaberrima</i>	African rice [40]	Yes	316	33,164
<i>Oryza sativa Indica</i>	Short grain rice [41]	Yes	427	40,745
<i>Oryza sativa Japonica</i>	Long grain rice [42]	Yes P	374	35,679
<i>Selaginella moellendorffii</i>	A model lycophyte genome and evolutionary reference point in the evolution of plants [43]	No	213	34,799
<i>Sorghum bicolor</i>	An economically important and widely grown cereal, particularly in Africa [44]	Yes	739	34,496
<i>Glycine max</i>	Soybean is an economically important crop, model legume, and one of the most important sources of animal feed protein and cooking oil [45]	Yes	973	54,174
<i>Solanum lycopersicum</i>	Tomato is an economically important food crop and a model for fruit ripening [46]	Yes P	782	34,675
Wheats	An economically important food crop, accounting for over 20 % of global agricultural production			
<i>Aegilops tauschii</i>	The diploid progenitor of the bread wheat D-genome [47]	No	3314	33,849
<i>Triticum aestivum</i>	Hexaploid bread wheat [48]	No	4460	108,569
<i>Triticum urartu</i>	The diploid progenitor of the bread wheat A-genome [49]	No	3747	34,843

The Chr/Pan column indicates whether or not the genome has been assembled into chromosomes (Yes or No) and if the species is included in the pan-taxonomic comparison (P). Note that the three wheat and four rice genomes are grouped together

Table 3

A list of the standard computational analyses that are routinely run over all genomes in Ensembl Plants

Pipeline name	Summary
Repeat feature annotation	Three repeat annotation tools are run, RepeatMasker (with species-specific repeat libraries), Dust, and TRF (<i>see</i> Fig. 2). http://ensemblgenomes.org/info/data/repeat_features
Noncoding RNA annotation	tRNAs and rRNAs are predicted using tRNAScan-SE and RNAmmer, respectively. Other ncRNA types are predicted by alignment to Rfam models (<i>see</i> Fig. 2). http://ensemblgenomes.org/info/data/ncrna
Feature density calculation	Feature density is calculated by chunking the genome into bins, and counting features of each type in each bin (<i>see</i> Fig. 1)
Annotation of external cross references	Database cross references are loaded from a predefined set of sources for each species, using either direct mappings or by sequence alignment. http://ensemblgenomes.org/info/data/cross_references
Ontology annotation	In addition to database cross references, ontology annotations are imported from external sources [18, 19]. Terms are additionally calculated using a standard pipeline [11]. http://ensemblgenomes.org/info/data/cross_references
Protein feature annotation	Translations are run through InterProScan [11] to provide protein domain feature annotations (<i>see</i> Fig. 5). http://ensemblgenomes.org/info/data/protein_features
Gene trees	The peptide comparative genomics (Compara) pipeline [17] computes feature rich gene trees for every protein in Ensembl Plants (<i>see</i> Fig. 4). http://ensemblgenomes.org/info/data/peptide_compara
Whole genome alignment	Whole genome alignments are provided for closely related pairs of species based on LastZ results. Where appropriate, Ka/Ks and synteny calculations are included. http://ensemblgenomes.org/info/data/whole_genome_alignment
Variation coding consequences	For those species with data for known variations, the coding consequences of those variations are computed for each protein-coding transcript [12]. http://plants.ensembl.org/info/docs/tools/vep/index.html

genotypes, population frequencies, linkage and statistical associations with phenotypes. A variety of views allow users to access this data and variant-centric warehouses are produced using BioMart. In addition, the Variant Effect Predictor tool allows users to upload their own data and see the functional consequence of self-reported variants on protein-coding genes [12]. In the case of the polyploid bread wheat genome, heterozygosity, inter-varietal variants and inter-homoeologous variants are all reported separately.

Table 4

List of some of the species-specific public datasets that are included in Ensembl Plants. There are two types of dataset, (1) sequence sets that are aligned to the genome (Fig. 2) and (2) variation datasets that are imported into the Ensembl variation architecture (Fig. 5). As the type and number of aligned datasets is large and continuously under revision, they will not be listed here. Instead see the homepage of the species of interest for an up-to-date list

List of variation datasets	
Species	Dataset
<i>Arabidopsis thaliana</i>	Several variation studies are included: (1) SNP identified from the screening of 1179 strains using the Affymetrix 250k Arabidopsis SNP chip and resequencing of 18 Arabidopsis lines, (2) variations from 392 strains from the 1001 Genomes Project [50]. Phenotype data has also been added from a GWAS study of 107 phenotypes in 95 inbred lines [51]
<i>Brachypodium distachyon</i>	Approximately 394,000 genetic variations have been identified by the alignment of transcriptome assemblies from three slender false brome (<i>Brachypodium sylvaticum</i>) populations [52]
<i>Hordeum vulgare</i>	Variations from two sources: (1) WGS survey sequence from four cultivars, Barke, Bowman, Igri, Haruna Nijo and a wild barley, <i>H. spontaneum</i> , (2) RNA-Seq was performed on the embryo tissues of nine spring barley varieties, Barke, Betzes, Bowman, Derkado, Intro, Optic, Quench, Sergeant, and Tocada [26]
<i>Oryza glaberrima</i>	Variation from two (unpublished) sources: (1) 20 diverse accessions of <i>Oryza glaberrima</i> and (2) 19 accessions of its wild progenitor, <i>Oryza barthii</i> , collected from geographically distributed regions of Africa
<i>Oryza sativa Indica</i>	Variations from two sources: (1) a collection of approximately four million SNPs based on a comparison of the Japonica and Indica genomes [53], (2) SNPs derived from the OMAP project based on alignments to <i>O. glaberrima</i> , <i>O. punctata</i> , <i>O. nivara</i> , and <i>O. rufipogon</i>
<i>Oryza sativa Japonica</i>	Variations covering four distinct studies: (1) a collection of approximately four million SNPs based on a comparison of the Japonica and Indica genomes [53], (2) SNPs derived from the OMAP project, (3) a SNP variation study involving 1311 SNPs across 395 accessions [54], and (4) OryzaSNP, a large-scale SNP variation study involving ~160 K SNPs in 20 diversity rice accessions [55]
<i>Sorghum bicolor</i>	Variations from a study of agroclimatic traits in the US sorghum association panel, comprising approximately 265,000 SNPs [56]
<i>Vitis vinifera</i>	SNPs identified by resequencing a collection of grape cultivars and wild Vitis species from the USDA germplasm collection [57]
<i>Zea mays</i>	Variations from HapMap2, incorporating 55 million SNPs and indels from 103 individuals [58]

2.2.4 Comparative Genomics

Two types of pairwise genome alignment are available in Ensembl Plants, generated using either LASTZ [13] (or its predecessor, BLASTZ [14]) or translated BLAT (tBLAT) [15], followed by downstream processing. LASTZ is typically used for closely related species, and tBLAT for more distant species. The method of

alignment affects the coverage of the genomes, with tBLAT expected to mostly find alignments in coding regions. The raw output from LASTZ or tBLAT comprises a pair of aligned sequences (a “block”); in the next step, non-overlapping, collinear sets of blocks are identified and a final step “nets” together compatible chains to find the best overall alignment for the reference species [16]. For highly similar species, an additional calculation defines high-level syntenic blocks between chromosomes. Alignment data is made available both graphically and for download, as described below.

The Ensembl Gene Tree pipeline [17] is used to calculate evolutionary relationships among members of protein families. Clusters of similar sequences are identified, then for each cluster several different programs are run, each of which attempts to reconcile the relationship between the sequences with the known evolutionary history. The TreeBeST program is used to construct a final consensus tree, which allows the identification of orthologues, paralogues, and, in the case of polyploid genomes, homoeologues. In addition to a plant-specific analysis, a number of plant genomes are included in a pan-taxonomic analysis, containing a representative selection of sequenced genomes from all domains of life which is constructed to show the relationships among members of widely conserved gene families.

3 Methods

There are many entry points and possible paths through the Ensembl Plants genome browser, supporting different use cases. Some common paths are presented below, with notes to indicate alternative paths and entry points. Although some details are necessarily omitted (*see Note 1*), following the instructions in the final section, Subheading 3.4, will allow a user to find more information on any of the topics discussed.

3.1 Browsing a Genome: Visually Integrating Public Datasets and User Data on a Common Reference

The Ensembl Plants browser allows users to navigate to a region of interest, configure the view to show specific features, attach their own data, and share the resulting view.

3.1.1 Navigating to a Species Homepage in Ensembl Plants

1. Navigate to <http://plants.ensembl.org>.
2. Select a species of interest from either the “Popular” shortlist, the “Select a species” drop down menu, or via the “View full list of all Ensembl Plants species” link (*see Notes 2–5*).

Chromosome summary

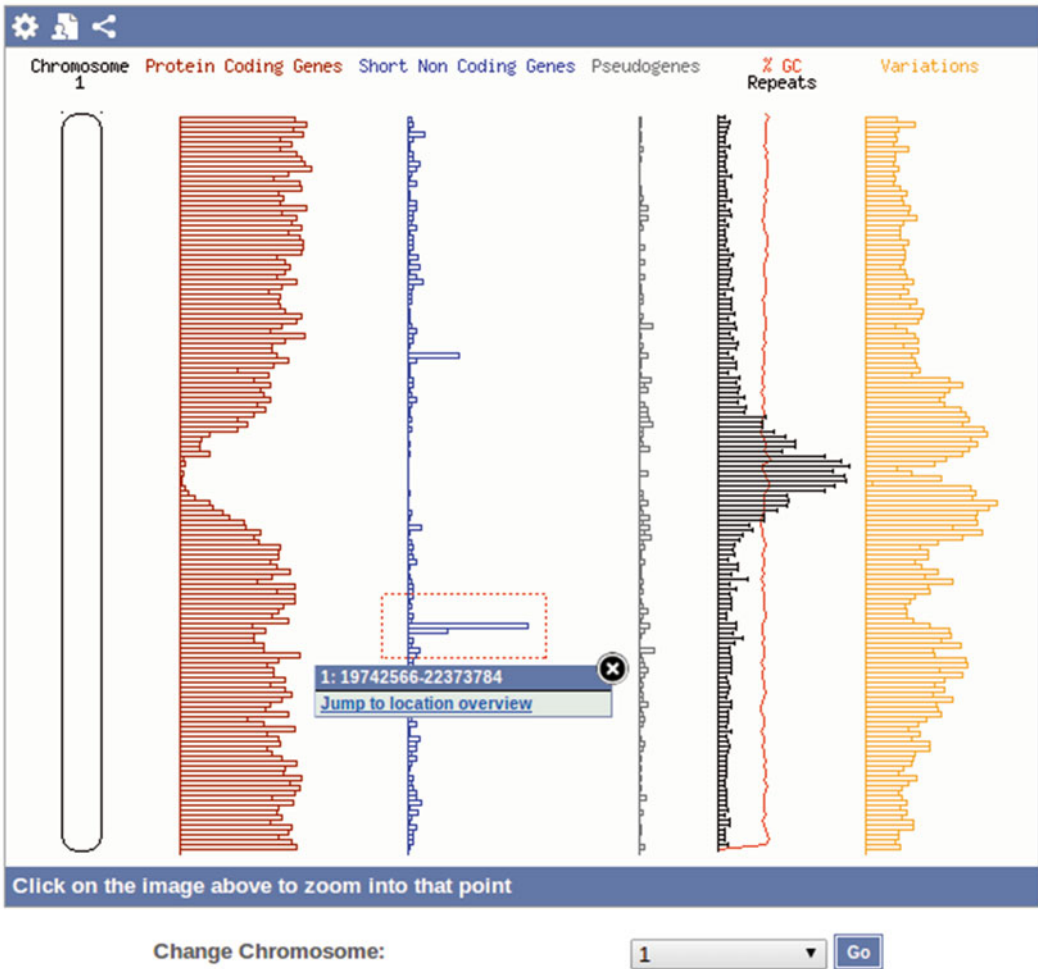


Fig. 1 The chromosome summary, shown here for *Arabidopsis thaliana* chromosome 1, gives a bird’s eye view of the chromosome structure, showing density histograms for genes, ncRNAs, pseudogenes, repeats, and variations. The GC ratio is plotted as a trend line on the repeat density histogram. A region of interest can be selected by clicking and dragging, allowing the user to jump in to the genome browser at a given chromosomal region

3.1.2 Enter the Genome Browser from a Chromosome Overview

1. On the species homepage, click the “View karyotype” icon (see **Notes 5** and **6**).
2. Click a chromosome and select “Chromosome summary” from the pop-up menu (see **Note 7**). The chromosome summary view (Fig. 1) gives a high-level, density-based overview of the organization of the plant’s chromosome.
3. Click and drag to select a small region of the chromosome and select “Jump to region overview” from the pop-up menu (see **Note 7**). The region overview is a configurable view showing

Region in detail 

Fig. 2 The upper “region overview” image shows a 200 kbp slice of chromosome 1 from *Arabidopsis thaliana*. Genes are color coded by type, protein coding, ncRNA, pseudogene, and “other”, in this case representing transposable elements. This high-level overview also includes blocks of synteny against rice and grape, with numbers indicating the syntenic chromosome. A 20 kbp window of the upper image is expanded in the lower “region in detail” image, showing tracks of various types, including an attached BAM file with expression data in Bur-O (*blue/grey*), precomputed EST alignments (*green*), gene models (colored by type), lncRNAs included via DAS, a set of small insertions from the 1001 genomes project (colored by transcript consequence), structural variations (*black and red*), and repeats (*grey*). The zoom widget between the two views can be used to control the lower view and the cog icon at the top left of each image can be used to configure the visible tracks and other display settings

selected sequence features for a large region of the genome, i.e., anything above 500 kbp (*see Figs. 2 and 3*).

4. For a more detailed view, allowing the full set of features to be displayed, select “Region in detail” from the left hand menu.
5. Zoom in using the “Drag/Select” option or the zoom widget (*see Fig. 2 and Note 8*).



Fig. 3 The track configuration dialogue for the region in detail view in Ensembl Plants. By default the active tracks are listed, allowing details to be viewed using the circular *i* icons on the *right*. Tracks are grouped into types in the *left hand menu*, allowing groups to be explored and activated in bulk. Tracks can be selected to show details of the genome sequence and assembly, gene model and variation datasets from the community, and precomputed sequence alignments including ESTs, RNA-Seq experiments, repeat features, oligo-probe, and marker sets. Tracks can be searched by name or description using the search box on the *top right*. Once a selection has been made, the user clicks the *arrow* on the *top right* to confirm and exit the dialogue

3.1.3 Configuring the Tracks and Features Shown on the Genome Browser

1. Click the configuration “cog” icon above the region in detail image to open the configuration menu for the image (*see* Fig. 2 and **Note 9**). The configuration menu shows the set of currently visible “active” tracks by default, with all available tracks categorized into the track menu on the left (*see* Fig. 3).
2. Tracks can be selected from the menu on the left and turned on or off individually or in groups (*see* **Notes 10–12**). Tracks are available that display genome sequence and assembly information, additional gene model and variation datasets, and precomputed sequence alignments including ESTs, RNA-Seq experiments, repeat features, oligo-probe, and marker sets (*see* Fig. 2). Some of this data is precomputed and hosted in Ensembl Plants, while other data is hosted on remote servers and loaded dynamically. Users can also configure the browser to load their own data.

3.1.4 Adding User-Supplied Data

1. Click the “Add your data” button in the left hand of the region in detail page (*see Note 13*).
2. A dialogue will ask you to name and specify the file format (data type) of your data. The site supports a number of different file formats for upload and visualization of data on the genome (Table 1), including sequence alignments, features, continuous-valued data, and variations.
3. After selecting a file format, the option to select a file from your computer, provide a URL, or paste in your data will appear (*see Notes 14 and 15*).
4. Click Upload, and follow the resulting link to see an example data point from your data, or simply click the tick mark (top right) and the browser image will redraw to include your newly added track.
5. Click the “Share this page” button under the left hand menu to generate a bookmark for your current configuration that can be shared.

3.2 Exploring a Gene: Sequence, Functional, Evolutionary, and Variation Analysis

Ensembl Plants allows users to search for a gene of interest, and display and download associated data, including transcript models, gene sequence, external database references, GO terms, protein domains, and gene trees. Variation data and associated variant-centric information can also be explored.

3.2.1 Finding a Gene of Interest

1. Search for a gene of interest on the Ensembl Plants homepage, e.g., “ARF” (*see Note 16*).
2. Pull up the gene summary page for a gene by clicking on its name in the search results (*see Note 17*). The gene summary page shows a localized image of the different transcripts of the gene depicting the UTR, exon, and CDS structure of the gene. The transcript table provides links and summary information for the alternative transcripts and gene products. Tabs at the top of the page can be used to switch between location-, gene-, and transcript-centric views of the selected gene. Various gene-centric views can be selected using the left hand menu (*see Note 18*) including pages for viewing sequence, function, and comparative information for the gene and, where available, associated variation, regulation, expression, literature, and phenotype information.

3.2.2 View and Download Gene Sequence

1. Select “Sequence” from the left hand menu. The gene sequence is shown with a configurable number of flanking bases. Exons of the selected gene are highlighted in bold red, while exons of any overlapping genes are highlighted in peach.
2. Click “Export data” in the left hand menu (*see Note 18*). Various export formats are available including FASTA and GFF3.

Specific options, such as soft or hard masking of repeats, can be configured for certain formats (*see Note 19*).

3. Select a transcript by clicking on the transcript ID in the transcript table at the top of the page and select “Exon”, “cDNA”, or “protein” under the “Sequence” section of the left hand menu.
4. Similar configuration and export options are available for each of these transcript-specific sequence views as for the gene sequence view.

3.2.3 View Database Cross References

1. Select “External references” from the left hand menu. External references link from the gene page in Ensembl Plants to the source database as well as several widely used databases for gene and/or protein information, including EntrezGene and UniProt.

3.2.4 Functional Annotation, Ontology Terms, and Functional Domains

1. Select “GO: biological process” from under the “Ontology” section of the left hand menu to see the biological process terms that have been associated with the gene from the Gene Ontology (GO) (*see Note 20*). A table gives details of each term annotated to the gene and information about how the annotation method.
2. Click the “Ancestry chart” tab to view the annotated terms in the context of the full ontology. Specific terms can be used to retrieve lists of annotated genes using BioMart, as described in Subheading 3.3.1. Similar views are available for terms annotated from other ontologies, including the other two domains of the Gene Ontology [18] and the three domains of the Plant Ontology [19].
3. Use the transcript table to select a protein translation for the gene by clicking on the protein ID. This will open the transcript tab on the protein summary view. The protein summary shows the predicted domain structure of the translation from InterPro [20], incorporating domain families from 11 separate databases.
4. Click a domain to bring up the pop-up menu. The pop-up menu links each domain back to the domain family in the source database.

3.2.5 Evolutionary Information

1. Select “Gene tree (image)” from under the “Plant Compara” section of the left hand menu of the gene tab (Fig. 4). The gene tree is the result of a phylogenetic analysis of the gene family to which the current gene belongs. The multiple sequence alignment of the family is shown schematically on the right, with the tree on the left. Collapsed branches of the tree are represented by colored “wedges” that summarize information within that part of the subtree (*see Note 21*).



Fig. 4 The gene tree shows the result of a phylogenetic analysis of the gene family to which the current gene (*red*) belongs. The multiple sequence alignment of the family is shown schematically on the *right*, with the tree on the *left*. Collapsed branches of the tree are represented by “wedges” that summarize information within that part of the subtree. A *blue node* indicates speciation, separating orthologues. A *red node* indicates duplication, separating paralogues. The tree can be colored by functional annotation, in this case, highlighting in *green* those genes that have been annotated with protein methyltransferase activity (GO:0008276), an enzyme that catalyzes the transfer of a methyl group (CH₃-) to a protein

2. Click on a “wedge” (or on its root node) to expand a branch using the pop-up menu.
3. Click on a branch node to see its underlying data, including the taxonomic range of the species within the node. Branch nodes are classified into speciation (blue) and duplication (red) indicating the most parsimonious evolutionary events consistent with the alignment and the known species taxonomy (*see Note 22*).
4. Click the name of a protein to jump to the associated transcript summary page for that protein in the given species.

3.2.6 Variation Information

1. Select “Variation image” from under the “Genetic Variation” section of the left hand menu (*see Note 23*). The image gives an overview of all the variations within the transcript in the context of the functional domains assigned to the protein (Fig. 5).

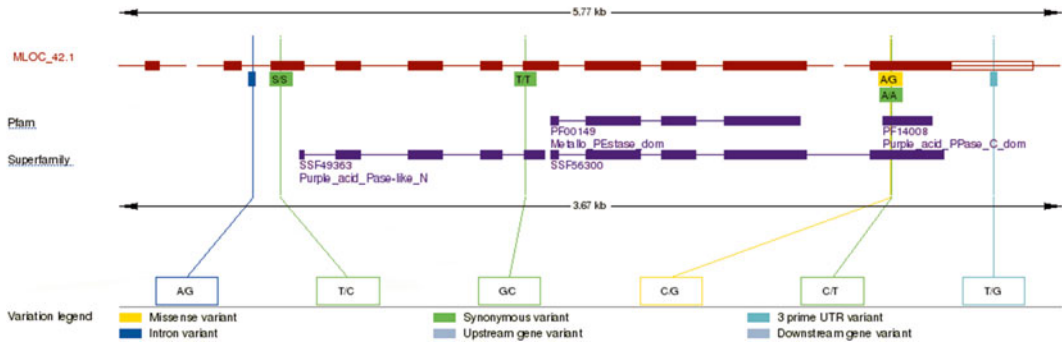


Fig. 5 The transcript variation image for the *Hordeum vulgare* MLOC_42.1 protein-coding transcript. The image gives an overview of all the variants within the transcript in the context of the functional domains assigned to the protein. *Upper boxes* highlight the amino acid change, where applicable, and *lower boxes* give the alleles. Variants are color coded according to their consequence type, missense, synonymous, and positional. A full list of consequence types is given here: http://www.ensembl.org/info/genome/variation/predicted_data.html. The transcripts, features, and variations can be clicked to explore more information about each object

2. Select “Variation table” from the left hand menu. A table of variations is shown, broken down by consequence type (*see Note 24*). Consequence types classify variations by the effects that each allele of the variation has on the transcript [12] using terms defined by the Sequence Ontology [21].
3. Click “Show” on one of the consequence types to get a detailed table of all variations within the transcript of that consequence type, e.g., missense variants.
4. Click on the ID of the variation in the detailed table to get to the variation-centric pages for that variation.
5. Click “Explore this variation” to access the various variation-centric pages for the selected variation.
6. Click the “Individual genotypes” icon to get the genotype of the variation in any associated samples.

3.3 Data Mining and Programmatic Access

There are several methods for bulk analysis of data in Ensembl Plants (*see Table 5*). These are illustrated with five examples: the use of the web-based BioMart data mining tool to identify all genes associated with a particular GO term and download the results as TSV; a Perl API script that retrieves a gene, its orthologues, and their GO terms; a REST API script to perform the same task; use of the FTP site to bulk download sequences and gene annotations; and direct connection to the Ensembl Genomes MySQL server.

3.3.1 Batch Retrieval of Genes Using Ontology Terms in BioMart

1. From <http://plants.ensembl.org>, click on the BioMart link in the top bar.
2. To search for genes, choose “Ensembl Plants Genes” from the first drop down menu, and then select the name of the

Table 5
A list of the different types of batch analysis available over the data in Ensembl Plants

Resource	Description
BioMart	A data mining tool for batch retrieval of gene-related data. Accessible via web interface and a language-independent REST API. http://ensemblgenomes.org/info/access/biomart
Perl API	A comprehensive Perl-based API for accessing all types of data available within Ensembl Plants. http://ensemblgenomes.org/info/access/api
REST Service	A language-independent API for retrieving data from Ensembl Plants. http://ensemblgenomes.org/info/access/rest
FTP download	Pre-generated genome-scale data files in a variety of commonly used formats (e.g., FASTA, GFF3, VCF). http://ensemblgenomes.org/info/access/ftp
Direct MySQL access	Public access to Ensembl Genomes MySQL databases. http://ensemblgenomes.org/info/access/mysql

species (and gene build) from the second drop down menu (*see Note 25*).

3. Click on “Filters” in the left hand menu to choose the criteria to use in your query (*see Note 26*).
4. To pick GO terms, expand the “Gene Ontology” filter, check “GO term accession”, and enter the GO term of interest (*see Fig. 6*).
5. Click on “Attributes” to choose what data to show in your results (*see Note 27*).
6. To show gene names and descriptions, expand the “Gene” attribute and check “Gene name” and “Gene description”. To show GO term details, scroll down, expand the “External” attribute, and check “GO term accession”, “GO term name”, and “GO term evidence code” (*see Fig. 6*).
7. To view results in the browser, click “Results”.
8. To download all results to your computer as a compressed tab separated file, select “Compressed file (.gz)” and “TSV” from the menus and click “Go”.

3.3.2 Retrieval of Genes and GO Annotation Using the Perl API

1. Install the Ensembl Perl API (*see Note 28*).
2. Load the Registry object with details of genomes available from the public Ensembl Genomes servers:

```
use warnings;
use strict;
use Bio::EnsEMBL::Registry;
```

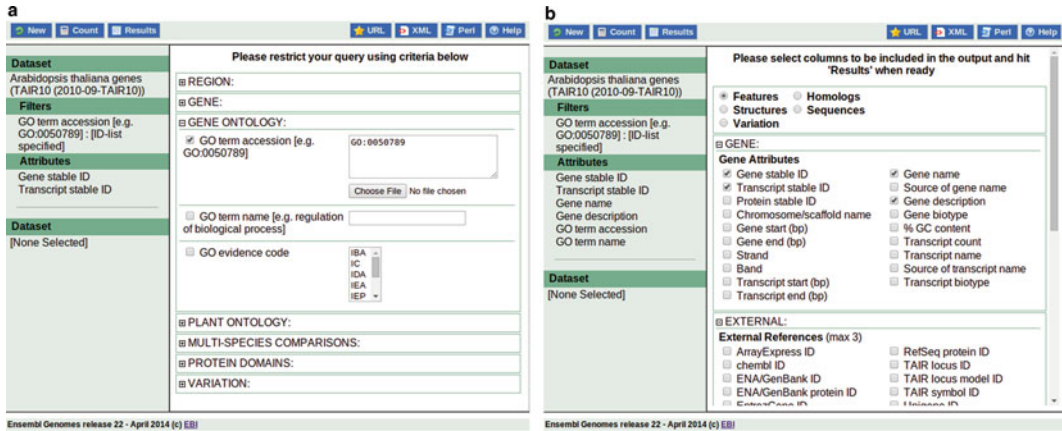


Fig. 6 Using BioMart to perform complex queries and retrieve data in bulk. The *left hand* image shows various filters that can be used to restrict the data returned. The *right hand* image shows the various attributes that can be selected in the results

```
Bio::Ensembl::Registry->
  load_registry_from_db(
    -USER => 'anonymous',
    -HOST => 'mysql.ebi.ac.uk',
    -PORT => '4157',
  );
```

3. Find the DEAR3 gene from *A. thaliana*:

```
# gene to look for
my $gene_name = 'DEAR3';

# species to look for
my $species = 'arabidopsis_thaliana';

# get a gene adaptor to work with genes from
the species
my $ga = Bio::Ensembl::Registry->
  get_adaptor($species, 'core', 'gene');

# find the gene with the specified name using
the adaptor
my ($gene_obj) =
  @{$gene_adaptor->
    fetch_all_by_external_name($gene_
      name)};
```

4. Find all orthologues from Tracheophytes in the plant Compara:

```
# compara database to search in
my $division = 'plants';

# get an adaptor to work with genes from compara
my $gene_member_adaptor = Bio::Ensembl::Registry->
  get_adaptor($division, 'compara', 'GeneMember');

# find the corresponding gene in compara
my $gene_member = $gene_member_adaptor->
  fetch_by_source_stable_id(
```

```
'ENSEMBLGENE',
$gene_obj->stable_id,
);
    # get an adaptor to work with homologues in compara
    my $homology_adaptor = Bio::EnsEMBL::Registry->
get_adaptor($division, 'compara', 'Homology');
    # find all homologues of the gene
    my @homologies =
@{$homology_adaptor->
fetch_all_by_Member($gene_member)};
    # filter out homologues based on taxonomy and type
    @homologies = grep {
$_->taxonomy_level eq 'Tracheophyta' &&
$_->description =~ m/ortholog/
} @homologies;
```

5. Find each orthologous protein:

```
    foreach my $homology (@homologies) {
# get the protein from the target
my $target = $homology->get_all_Members->[1];
my $translation = $target->get_Translation;
print
$target->genome_db->name, ' orthologue ',
$translation->stable_id, "\n";
}

```
6. For each translation, print information about GO annotation:

```
    # find all the GO terms for this translation
    foreach my $go_term (@{$translation->
get_all_DBEntries('GO')}) {
# print some information about each GO annotation
print
$go_term->primary_id, ' ', $go_term->description;
# print the evidence for the GO annotation
print
' Evidence: ', (join ' ', map {$_->[0]}
@{$go_term->get_all_linkage_info}), "\n";
}

```

3.3.3 Retrieval of Genes and GO Annotation Using the REST API

1. Create an HTTP client and a helper function for invoking a REST endpoint:

```
    use strict;
    use warnings;
    use JSON;
    use HTTP::Tiny;
    use Data::Dumper;
    # create an HTTP client
    my $http = HTTP::Tiny->new;
    my $server = 'http://beta.rest.ensemblgenomes.org';
```



```

        # function for invoking endpoint
        sub call_endpoint {
my ($url) = @_;
print "Invoking $url\n";
my $response = $http->
get($url, {headers =>
{'Content-type' => 'application/json'}
});
return decode_json($response->{content});
}

```

2. Find homologues of *A. thaliana* DEAR3 gene:

```

        # find homologues of A. thaliana DEAR3 gene
my $gene = 'DEAR3';
my $species = 'arabidopsis_thaliana';
my $division = 'plants';
my $url =
join("/", $server, 'homology/symbol', $species, $gene).
"?content-type=application/json&$compara=$division";
        # call url endpoint and get a hash back
my $homologue_data = call_endpoint($url);
        # parse the homologue list from the response
        my @homologies = @{$homologue_data->{data}[0]
{homologies}};
        # filter out homologues based on taxonomy and type
        @homologies = grep {
$_->taxonomy_level eq 'Tracheophyta' &&
$_->description =~ m/ortholog/
} @homologies;

```
3. Print some information about the orthologous protein:

```

        for my $homologue (@homologies) {
my $target_species = $homologue->{target}{species};
my $target_id = $homologue->{target}{protein_id};
print "$target_species orthologue $target_id\n";
}

```
4. For each translation, print information about GO annotation using the xrefs/id endpoint:

```

my $url =
join("/", $server, xrefs/id).
"?content-type=application/json;external_
db=GO;all_levels=1";
my $go_data = call_endpoint($url);
for my $go (@{$go_data}) {
print
$go->{display_id}, ' ', $go->{description},
' Evidence: ', join(' ', @{$go->{linkage_types}}),
"\n";
}

```

3.3.4 Retrieval of All Peptide Sequences Using FTP

1. Navigate to <http://plants.ensembl.org/> and click on “Downloads” in the top bar.
2. From the right-most box (entitled “Download databases & software”), click “Download data via FTP”.
3. Downloads are grouped by species in alphabetical order in the main table. To find your species of interest, either navigate through the table page by page, or type the name of the species into the “Filter” box in the header of the table.
4. For a given species, click on “FASTA (protein)” to go to the FTP directory containing peptide data in FASTA format. The file with the extension “.pep.all.fa.gz” contains all peptide sequences for that species (*see Note 29*).

3.3.5 Direct Access to MySQL

1. Use your MySQL client to connect to host “mysql.ebi.ac.uk”, port 4157 as the user “anonymous,” e.g., `mysql --user anonymous --port 4157 --host mysql.ebi.ac.uk`
2. Databases are named for the relevant Ensembl and Ensembl Genomes releases, e.g., `arabidopsis_thaliana_core_22_75_10` comes from release 22 of Ensembl Genomes, using version 75 of the Ensembl platform and based on release 10 of the TAIR assembly and annotation.
3. The schemas for different Ensembl databases are described in <http://www.ensembl.org/info/docs/api/index.html>.

3.4 Learning More and Getting Help

Overall help and documentation for the website including FAQs, tutorials, and detailed information about the project, datasets, and pipelines that we run can be found under the “Help” and “Documentation” links at the top of every page. Context-sensitive help for specific views can be found under the circular *i* icons that appear next to the page headers. Details of specific datasets can be found in the infobox for each track in the browser or configuration pages. Detailed information for each species can be found on the species homepage. If the available documentation cannot answer your question, a helpdesk is provided (mail helpdesk@ensemblgenomes.org with your query).

The following list of pages can be used as a starting point for learning more about the Ensembl browser.

There are various “Train online” resources related to Ensembl and Ensembl Genomes:

- <http://www.ebi.ac.uk/training/online/course/ensembl-genomes-non-chordates-quick-tour>
– The Ensembl Genomes Quick Tour.
- <http://www.ebi.ac.uk/training/online/course/ensembl-browsing-chordate-genomes>

- <http://www.ebi.ac.uk/training/online/course/ensembl-filmed-browser-workshop>
 - Two Ensembl browsing courses.
- <http://www.ebi.ac.uk/training/online/course/ensembl-filmed-api-workshop>
 - The API training course.And additional online documentation:
- <http://www.ensembl.org/info/website/index.html>
 - A starting point for information about using the website
- <http://www.ensembl.org/info/website/tutorials/index.html>
 - A list of Ensembl tutorials and worked examples
- <http://www.ensembl.org/info/website/upload/index.html>
 - Clips and documentation focused on adding custom tracks to Ensembl
- http://www.ensembl.org/info/website/control_panel.html
 - All about the Ensembl control panel (referred to here as the configuration menu).
- <http://www.ensembl.org/info/website/glossary.html>
 - A glossary of terms used in the browser.

4 Notes

1. For example, the methods here don't cover the BLAST or Sequence Search entry points, nor any of the dedicated Ensembl "Tools" such as the Assembly converter, Region Report, or Variant Effect Predictor. *See* <http://plants.ensembl.org/tools.html>.
2. You can login to customize the list of "popular" genomes shown on the Ensembl Plants homepage.
3. The species drop-down menu is grouped into broad taxonomic levels.
4. The full list of species also shows which types of data are available for each species. Use the key to see which species has a variation, comparative, and alignment data (typically EST or RNA-Seq).
5. Icons are used on the species homepage to link in to the genome browser and its associated gene- and transcript-centric pages.
6. The karyotype icon is only available for genomes with chromosome-scale assemblies (*see* Table 2 for the full list of genomes and their assembly status).

7. The pop-up menus provide context-sensitive information and links for the sequence features in the browser. The menu will typically pop-up when clicking features or clicking and dragging on the browser image.
8. The detail pane will show when the region selected is less than or equal to between 200 and 500 kbp, depending on the species.
9. Any image can be configured by clicking the configuration “cog” icon above it. Alternatively, all the configurable items on a page can be configured from a single “tabbed” menu by selecting the “Configure this page” button under the left hand menu.
10. Depending on the feature type different visualization styles may be selected, such as having labeled or unlabeled features or whether or not to collapse overlapping features. For the full list of available styles, *see* <http://www.ensembl.org/Help/Faq?id=335>.
11. Tracks can be rapidly searched using the track name or description by using the “Find a track” search box in the upper right of the config menu (*see* Fig. 3).
12. Information about each track is available by clicking the circular *i* icon to the right of each track.
13. This button will change from “Add your data” to “Manage your data” once any data has been added.
14. You are allowed to upload smaller files (up to 5 MB). Larger data files may be attached by URL.
15. Attached files may require an additional index file (*see* Table 1 for details).
16. By default, the search on the Ensembl Plants homepage will return matches to genes across all species. You can select a specific species to search against before searching, or filter the results by species after searching.
17. The gene summary page may also be accessed from the genome browser by clicking on a gene and clicking the gene identifier in the pop-up menu.
18. The left hand menu changes to provide different options on the location, gene, and transcript views.
19. Sequence can be exported in HTML, text, or compressed text format.
20. Functional annotations from the Gene Ontology (GO) and the Plant Ontology (PO) are attached to genes, transcripts, and translations from various sources. For more details, *see* http://ensemblgenomes.org/info/data/cross_references.

21. The tree can be highlighted by functional annotation using the table of annotations above the tree. Selecting an annotation in the table will highlight the branches of the tree where that annotation has been applied (*see* Fig. 4).
22. Tables of orthologues, paralogues, and, where appropriate, homoeologues are available from options in the left hand menu.
23. If variation data has not been made available for the selected species (*see* Table 4), the variation options will be greyed out. In this case you can attach variation data to the reference in VCF format (*see* Subheading 3.1.4) and perform VEP analysis using the VEP tool (<http://plants.ensembl.org/tools.html>).
24. The color coding used in the table is the same as that used in the region view of the genome browser (*see* Fig. 2) and the variation image (*see* Fig. 5). The complete list of consequence types is given here: http://www.ensembl.org/info/genome/variation/predicted_data.html
25. Alternatively, choose “Ensembl Plants Variation” to query over variation datasets.
26. Filters are available for genomic region, gene attributes, ontology terms, comparative genomics, functional domains, and variation types.
27. There are five broad classes of attributes to choose from, Features (used in the example), Homologs (to select data from gene trees), Structures (to obtain gene structure information), Sequences (for various DNA or peptide sequences), and Variation (for variation data).
28. Instructions for installing the Ensembl Perl API can be found here: http://www.ensembl.org/info/docs/api/api_installation.html
29. Direct FTP access is also possible from <ftp://ftp.ensembl.org/pub/current/plants> with data organized by type and species. For instance, *A. thaliana* FASTA sequence is available from ftp://ftp.ensemblgenomes.org/pub/current/plants/fasta/arabidopsis_lyrata/pep/

Acknowledgements

“The transPLANT project is funded by the European Commission within its 7th Framework Programme, under the thematic area “Infrastructures”, contract number 283496”.

References

1. Ribaut JM, Hoisington D (1998) Marker-assisted selection: new tools and strategies. *Trends Plant Sci* 3(6):236–239
2. Goddard ME, Hayes BJ (2007) Genomic selection. *J Anim Breed Genet* 124(6):323–330
3. Rafalski JA (2010) Association genetics in crop improvement. *Curr Opin Plant Biol* 13(2):174–180
4. Kleinhofs A, Behki R (1977) Prospects for plant genome modification by nonconventional methods. *Annu Rev Genet* 11(1):79–101
5. Hartung F, Schiemann J (2014) Precise plant breeding using new genome editing techniques: opportunities, safety and regulation in the EU. *Plant J* 78(5):742–752
6. http://wikipedia.org/wiki/List_of_sequenced_plant_genomes. 2012
7. <http://faostat.fao.org/>. 2011
8. Kersey PJ, Allen JE, Christensen M et al (2014) Ensembl genomes 2013: scaling up access to genome-wide data. *Nucleic Acids Res* 42(D1):D546–D552
9. Monaco MK, Stein J, Naithani S et al (2014) Gramene 2013: comparative plant genomics resources. *Nucleic Acids Res* 42(D1):D1193–D1199
10. Kasprzyk A (2011) BioMart: driving a paradigm change in biological data management. *Database* 2011:bar049
11. Jones P, Binns D, Chang H-Y et al (2014) InterProScan 5: genome-scale protein function classification. *Bioinformatics* 30(9):1236–1240
12. McLaren W, Pritchard B, Rios D et al (2010) Deriving the consequences of genomic variants with the Ensembl API and SNP effect predictor. *Bioinformatics* 26(16):2069–2070
13. Harris RS (2007) Improved pairwise alignment of genomic DNA. *ProQuest*, Ann Arbor, p 84
14. Schwartz S, James Kent W, Smit A et al (2003) Human–mouse alignments with BLASTZ. *Genome Res* 13(1):103–107
15. Kent WJ (2002) BLAT—the BLAST-like alignment tool. *Genome Res* 12(4):656–664
16. Kent WJ, Baertsch R, Hinrichs A, Miller W, Haussler D (2003) Evolution’s cauldron: duplication, deletion, and rearrangement in the mouse and human genomes. *Proc Natl Acad Sci* 100(20):11484–11489
17. Vilella AJ, Severin J, Ureta-Vidal A et al (2009) EnsemblCompara GeneTrees: complete, duplication-aware phylogenetic trees in vertebrates. *Genome Res* 19(2):327–335
18. Ashburner M, Ball CA, Blake JA et al (2000) Gene ontology: tool for the unification of biology. *Nat Genet* 25(1):25–29
19. Cooper L, Ramona L, Walls JE et al (2013) The plant ontology as a tool for comparative plant anatomy and genomic analyses. *Plant Cell Physiol* 54(2):e1
20. Burge S, Kelly E, Lonsdale D et al (2012) Manual GO annotation of predictive protein signatures: the InterPro approach to GO curation. *Database* 2012:bar068
21. Eilbeck K, Lewis SE, Mungall CJ et al (2005) The Sequence Ontology: a tool for the unification of genome annotations. *Genome Biol* 6(5):R44
22. Chamala S, Chanderbali AS, Der JP et al (2013) Assembly and validation of the genome of the nonmodel basal angiosperm *Amborella*. *Science* 342(6165):1516–1517
23. Hu TT, Pattyn P, Bakker EG et al (2011) The *Arabidopsis lyrata* genome sequence and the basis of rapid genome size change. *Nat Genet* 43(5):476–481
24. Arabidopsis Genome Initiative (2000) Analysis of the genome sequence of the flowering plant *Arabidopsis thaliana*. *Nature* 408(6814):796
25. D’Hont A, Denoeud F, Aury JM et al (2012) The banana (*Musa acuminata*) genome and the evolution of monocotyledonous plants. *Nature* 488(7410):213–217
26. International Barley Genome Sequencing Consortium (2012) A physical, genetic and functional sequence assembly of the barley genome. *Nature* 491(7426):711–716
27. International Brachypodium Initiative (2010) Genome sequencing and analysis of the model grass *Brachypodium distachyon*. *Nature* 463(7282):763–768
28. Brassica rapa Genome Sequencing Project Consortium (2011) The genome of the mesopolyploid crop species *Brassica rapa*. *Nat Genet* 43(10):1035–1039
29. Merchant SS, Prochnik SE, Vallon O et al (2007) The *Chlamydomonas* genome reveals the evolution of key animal and plant functions. *Science* 318(5848):245–250
30. Matsuzaki M, Misumi O, Shin-I T et al (2004) Genome sequence of the ultrasmall unicellular

- red alga *Cyanidioschyzon merolae* 10D. *Nature* 428(6983):653–657
31. Consortium for Grapevine Genome Characterization, (2007) The grapevine genome sequence suggests ancestral hexaploidization in major angiosperm phyla. *Nature* 449(7161):463–467
 32. Schnable PS, Ware D, Fulton RS et al (2009) The B73 maize genome: complexity, diversity, and dynamics. *Science* 326(5956):1112–1115
 33. Young ND, DeBellé F, Oldroyd GE et al (2011) The Medicago genome provides insight into the evolution of rhizobial symbiosis. *Nature* 480(7378):520–524
 34. Bennetzen JL, Schmutz J, Wang H et al (2012) Reference genome sequence of the model plant *Setaria*. *Nat Biotechnol* 30(6):555–561
 35. Rensing SA, Lang D, Zimmer AD et al (2008) The Physcomitrella genome reveals evolutionary insights into the conquest of land by plants. *Science* 319(5859):64–69
 36. Tuskan GA, Difazio S, Jansson S et al (2006) The genome of black cottonwood, *Populus trichocarpa* (Torr. & Gray). *Science* 313(5793):1596–1604
 37. Potato Genome Sequencing Consortium (2011) Genome sequence and analysis of the tuber crop potato. *Nature* 475(7355):189–195
 38. International Peach Genome Initiative (2013) The high-quality draft genome of peach (*Prunus persica*) identifies unique patterns of genetic diversity, domestication and genome evolution. *Nat Genet* 45(5):487–494
 39. Chen J, Huang Q, Gao D et al (2013) Whole-genome sequencing of *Oryza brachyantha* reveals mechanisms underlying *Oryza* genome evolution. *Nat Commun* 4:1595
 40. http://plants.ensembl.org/Oryza_glaberrima/Info/Annotation/
 41. Yu J, Hu S, Wang J et al (2002) A draft sequence of the rice genome (*Oryza sativa* L. ssp. *indica*). *Science* 296(5565):79–92
 42. International Rice Genome Sequencing Project (2005) The map-based sequence of the rice genome. *Nature* 436(7052):793–800
 43. Banks JA, Nishiyama T, Hasebe M et al (2011) The Selaginella genome identifies genetic changes associated with the evolution of vascular plants. *Science* 332(6032):960–963
 44. Paterson AH, Bowers JE, Bruggmann R et al (2009) The Sorghum bicolor genome and the diversification of grasses. *Nature* 457(7229):551–556
 45. Schmutz J, Cannon SB, Schlueter J et al (2010) Genome sequence of the palaeopolyploid soybean. *Nature* 463(7278):178–183
 46. Tomato Genome Consortium (2012) The tomato genome sequence provides insights into fleshy fruit evolution. *Nature* 485(7400):635–641
 47. Jia J, Zhao S, Kong X et al (2013) Aegilops tauschii draft genome sequence reveals a gene repertoire for wheat adaptation. *Nature* 496(7443):91–95
 48. http://plants.ensembl.org/Triticum_aestivum/Info/Annotation/
 49. Ling HQ, Zhao S, Liu D et al (2013) Draft genome of the wheat A-genome progenitor *Triticum urartu*. *Nature* 496(7443):87–90
 50. Clark RM, Schweikert G, Toomajian C et al (2007) Common sequence polymorphisms shaping genetic diversity in *Arabidopsis thaliana*. *Science* 317(5836):338–342
 51. Atwell S, Huang YS, Vilhjálmsson BJ et al (2010) Genome-wide association study of 107 phenotypes in *Arabidopsis thaliana* inbred lines. *Nature* 465(7298):627–631
 52. Fox SE, Preece J, Kimbrel JA et al (2013) Sequencing and de novo transcriptome assembly of *Brachypodium sylvaticum* (Poaceae). *Appl Plant Sci* 1(3)
 53. Yu J, Wang J, Lin W et al (2005) The genomes of *Oryza sativa*: a history of duplications. *PLoS Biol* 3(2), e38
 54. Zhao K, Wright M, Kimball J et al (2010) Genomic diversity and introgression in *O. sativa* reveal the impact of domestication and breeding on the rice genome. *PLoS One* 5(5):e10780
 55. McNally KL, Childs KL, Bohnert R et al (2009) Genomewide SNP variation reveals relationships among landraces and modern varieties of rice. *Proc Natl Acad Sci* 106(30):12273–12278
 56. Morris GP, Ramu P, Deshpande SP et al (2013) Population genomic and genome-wide association studies of agroclimatic traits in sorghum. *Proc Natl Acad Sci* 110(2):453–458
 57. Myles S, Chia J-M, Hurwitz B et al (2010) Rapid genomic characterization of the genus *vitis*. *PLoS One* 5(1), e8219
 58. Chia JM, Song C, Bradbury PJ et al (2012) Maize HapMap2 identifies extant variation from a genome in flux. *Nat Genet* 44(7):803–807

Gramene: A Resource for Comparative Analysis of Plants Genomes and Pathways

Marcela Karey Tello-Ruiz, Joshua Stein, Sharon Wei,
Ken Youens-Clark, Pankaj Jaiswal, and Doreen Ware

Abstract

Gramene is an integrated informatics resource for accessing, visualizing, and comparing plant genomes and biological pathways. Originally targeting grasses, Gramene has grown to host annotations for economically important and research model crops, including wheat, potato, tomato, banana, grape, poplar, and *Chlamydomonas*. Its strength derives from the application of a phylogenetic framework for genome comparison and the use of ontologies to integrate structural and functional annotation data. This chapter outlines system requirements for end users and database hosting, data types and basic navigation within Gramene, and provides examples of how to (1) view a phylogenetic tree for a family of transcription factors, (2) explore genetic variation in the orthologues of a gene with a known trait association, and (3) upload, visualize, and privately share end user data into a new genome browser track.

Moreover, this is the first publication describing Gramene's new web interface—intended to provide a simplified portal to the most complete and up-to-date set of plant genome and pathway annotations.

Key words Plant genome, Reference genomes, Comparative genomics, Phylogenetics, Gene homology, Synteny, Genetic variation, Structural variation, Plant pathways

1 Introduction

Gramene (<http://www.gramene.org>; Fig. 1) is a curated online resource for comparative functional genomics in socioeconomically important crops and research model plant species, currently hosting over 30 completely sequenced plant reference genomes (Table 1; [1–31]). Each plant genome features community-based gene annotations provided by primary sources and enriched with supplemental annotations from cross-referenced sources, functional classification, and comparative phylogenomics analysis performed in-house. Increasing amounts of genetic and structural variation data derived both from data repositories and through collaboration with large-scale resequencing and genotyping initiatives are also available for visualization and analysis (Table 2; [32–40]).

Gramene: A comparative resource for plants | Gramene

Gramene: A comparative resour...

www.gramene.org

Google

Search

Home Release Notes Collaborators Contact About Archive Citing Gramene Outreach

Navigation


- Current Release (40)
- Search
- Genomes
- Pathways
- BLAST
- Gramene Mart
- News
- Archive (Build 39)
- Download
- Web Services
- Contact
- Tools

Recent blog posts

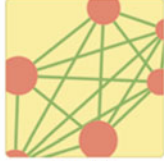
- Gramene at the Maize Genetics Conference 2014
- PAG 2014: Some Highlights
- Gramene workshop at the Maize Genetics Conference 2014 — See you in Beijing!
- Tenure-track faculty position in Plant-Microbe Interactions at Oregon State University
- ABA Biosynthesis and Signaling

More

Gramene: A comparative resource for plants



Genomes



Pathways

Gramene is a curated, open-source, integrated data resource for comparative functional genomics in crops and model plant species. Our goal is to facilitate the study of cross-species comparisons using information generated from projects supported by public funds. Gramene currently hosts annotated whole genomes in over two dozen plant species and partial assemblies for almost a dozen wild rice species in the Ensembl browser, genetic and physical maps with genes, ESTs and QTLs locations, genetic diversity data sets, structure-function analysis of proteins, plant pathways databases (BioCyc and Plant Reactome platforms), and descriptions of phenotypic traits and mutations.

Gramene Portals

- Genome Browser:** Browse gene annotations & diversity data
- BLAST:** Align DNA & protein sequences
- Plant Reactome:** Browse metabolic & regulatory pathways
- Pathways databases:** BioCyc based cellular metabolic networks for 10 plant species
- Gramene Mart:** Customized data queries
- Bulk downloads**
- ARCHIVE - Markers, Proteins and Ontology databases, QTLs, Comparative Maps**

Fig. 1 Gramene's homepage

Furthermore, plant pathway databases generated by applying both manual curation and automated methods complement available sequence-based gene annotations. Most advantageous to plant researchers and bioinformaticians is that, by using a core set of consistently applied protocols, Gramene offers a reference resource for basic and translational research in plants.

Gramene is driven by several platform infrastructures or modules that are linked to provide a unified user experience. Its Genome Browser (<http://ensembl.gramene.org>; Fig. 2) takes advantage of the Ensembl project's infrastructure [41] to provide an interface for exploring genome features, functional ontologies, variation

Table 1
Plant reference genome sequences in Gramene build 41 (May 2014)

Species	Reference genome status	Assembly/gene space annotation	Literature references
<i>Aegilops tauschii</i> (goatgrass, wheat D-genome progenitor)	Complete draft	GCA 000347335.1/2013-12-BGI	Jia et al. [1]
<i>Amborella trichopoda</i>	Complete draft	AMTRI.0 (GCA_000471905.1)/2014-01-AGD	Amborella Genome Project [2]; Chamala et al. [3]
<i>Arabidopsis lyrata</i>	Complete draft	v.1.0/2008-12-Araly1.0	Hu et al. [4]
<i>Arabidopsis thaliana</i>	Complete draft	TAIR10/2010-09-TAIR10	Arabidopsis Genome Initiative [5]
<i>Brachypodium distachyon</i>	Complete draft	v1.0/2010-02-Brachy1.2	The International Brachypodium Initiative [6]
<i>Brassica rapa</i> (Chinese cabbage)	Complete draft	IVFCAASv1/bra_v1.01_SP2010_01	Wang et al. [7]
<i>Chlamydomonas reinhardtii</i> (green algae)	Complete draft	v3.1 (GCA_000002595.2)/2007-11-ENA	Merchant et al. [8]
<i>Cyanidioschyzon merolae</i> (red algae)	Complete draft	ASM9120v1/2008-11-ENA	Matsuzaki et al. [9]
<i>Glycine max</i> (soybean)	Complete draft	V1.0 (GCA_000004515.1)/JGI-Glyma-1.1	Schmutz et al. [10]
<i>Hordeum vulgare</i> (barley)	Complete draft	030312v2/IBSC 1.0	The International Barley Genome Sequencing Consortium [11]
<i>Medicago truncatula</i>	Complete draft	GCA 000219495.1/2011-11-EnsemblPlants	Young et al. [12]
<i>Musa acuminata</i> (banana)	Complete draft	MA1/2012-08-Cirad	D'Hont et al. [13]
<i>Oryza barthii</i>	Complete draft	O.barthii_v1 (GCA_000182155.2)/OGE-MAKER	OGE/OMAP (NSF award #1026200)
<i>Oryza brachyantha</i>	Complete draft	Oryza brachyantha.v1.4b (GCA_000231095.2)/OGE-MAKER	Chen et al. [14]
<i>Oryza glaberrima</i>	Complete draft	AGII.1/2011-05-AGI	OGE/OMAP (NSF award #1026200)
<i>Oryza glumaepatula</i>	Complete draft	Oryza glumaepatula_v1.5 (GCA_000576495.1)/OGE-MAKER	OGE/OMAP (NSF award #1026200)

(continued)

Table 1
(continued)

Species	Reference genome status	Assembly/gene space annotation	Literature references
<i>Oryza meridionalis</i>	Complete draft	Oryza_meridionalis_v1.3 (GCA_000338895.2)/OGE-MAKER	OGE/OMAP (NSF award #1026200)
<i>Oryza nivara</i>	Complete draft	Oryza_nivara_v1.0 (GCA_000576065.1)/OGE-MAKER	OGE/OMAP (NSF award #1026200)
<i>Oryza punctata</i>	Complete draft	Oryza_punctata_v1.2 (GCA_000573905.1)/OGE-MAKER	OGE/OMAP (NSF award #1026200)
<i>Oryza sativa ssp. indica</i>	Complete draft	ASM465v1 (GCA_000004655.2)/2010-07-BGI	Yu et al. [15]; Zhao et al. [16]
<i>Oryza sativa ssp. japonica</i> (rice)	Complete draft	IRGSP-1.0/MSU7 (IRGSP-1.0)	International Rice Genome Sequencing [17]; Kawahara et al. [18]
<i>Physcomitrella patens</i> (moss)	Complete draft	ASM242v1/2011-03-Phypal.6	Rensing et al. [19]
<i>Populus trichocarpa</i> (poplar)	Complete draft	JGI 2.0/2010-01-JGI	Tuskan et al. [20]
<i>Prunus persica</i> (peach)	Complete draft	Prupe1_0 (GCA_000346465.1)/2013-03	Verde et al. [21]
<i>Selaginella moellendorffii</i> (spikemoss)	Complete draft	v1.0/2011-05-ENA	Banks et al. [22]
<i>Setaria italica</i> (foxtail millet)	Complete draft	JGIv2.0/JGIv2.1	Bennetzen et al. [23]; Zhang et al. [24]
<i>Solanum lycopersicum</i> (tomato)	Complete draft	SL2.40/ITAG2.3	Tomato Genome Consortium [25]
<i>Solanum tuberosum</i> (potato)	Complete draft	3.0/SolTub 3.0	Potato Genome Sequencing Consortium [26]
<i>Sorghum bicolor</i>	Complete draft	Sorbi1/2007-12-JGI (Sbi1.4)	Paterson et al. [27]
<i>Triticum aestivum</i> (bread wheat)	Complete chromosome survey	IWGSP1/MIPS2.1	International Barley Genome Sequencing Consortium et al. [11]; Brenchley et al. [28]

(continued)

Table 1
(continued)

Species	Reference genome status	Assembly/gene space annotation	Literature references
<i>Triticum urartu</i> (einkorn wheat, A-genome progenitor)	Complete draft	ASM34745v1 (GCA 000347455.1)/2013-04-BGI	Ling et al. [29]
<i>Vitis vinifera</i> (grape)	Complete draft	IGGP_12X/2012-07-CRIBI	Jaillon et al. [30]; Myles et al. [31]
<i>Zea mays</i> (corn)	Complete draft	B73 RefGen AGPv3/5b+	Schnable et al. [31]; Wei et al. [31]
<i>Leersia perrieri</i> (chr. 3s)	Partial	454.pools.2012Feb/2012-10-CSHL	OGE/OMAP (NSF award #1026200)
<i>Oryza granulata</i> (chr. 3s)	Partial	454.pools.2012Feb/2012-10-CSHL	OGE/OMAP (NSF award #1026200)
<i>Oryza longistaminata</i> (chr. 3s)	Partial	OGE.2012Jul/2012-10-CSHL	OGE/OMAP (NSF award #1026200)
<i>Oryza minuta BB</i> (chr. 3s)	Partial	BAC.Sanger.1.1 (May 2011)/ CSHLv3.1	OGE/OMAP (NSF award #1026200)
<i>Oryza minuta CC</i> (chr. 3s)	Partial	BAC.Sanger.1.1 (May 2011)/ CSHLv3.1	OGE/OMAP (NSF award #1026200)
<i>Oryza officinalis</i> (chr. 3s)	Partial	BAC.Sanger.1.1 (May 2011)/ CSHLv3.1	OGE/OMAP (NSF award #1026200)
<i>Oryza rufipogon</i> (chr. 3s)	Partial	454.pools.1.1 (Jul 2010)/CSHL	OGE/OMAP (NSF award #1026200)

data, and comparative phylogenomics. Since 2009 Gramene has partnered with the Plants division of Ensembl Genomes [42] to jointly produce this resource, each benefiting from the other's proximity to research communities in the USA and Europe, respectively. This collaboration has also facilitated timely adoption of innovative tools and software updates that accompany frequent version releases by the Ensembl project [41].

Since the last edition of this volume in 2007 [43], Gramene has also become a portal for pathway databases developed and curated internally or mirrored from external sources. Two pathway platforms are currently supported: (1) Gramene's Pathway Tools (<http://pathway.gramene.org>; Fig. 3) to emphasize the annotation of metabolic and transport pathways [44–46], and (2) Plant Reactome (<http://plantreactome.gramene.org>; Fig. 4) to facilitate the annotation of metabolic and regulatory pathways. The Pathway Tools

Table 2
Genetic and structural variation data in Gramene build 41 (May 2014)

Species	Variants	Source	Studies
<i>Arabidopsis thaliana</i>	14,234,197 SV: 13,667	250K SNPs×1179 accessions 1001 genomes project: 411 resequenced accessions	Atwell et al. [32]
<i>Brachypodium distachyon</i>	327,988	394K SNPs×3 accessions of <i>Brachypodium sylvaticum</i>	Fox et al. [33]
<i>Oryza glaberrima</i>	7,704,409	Resequenced 20 accessions & 19 accessions of its wild progenitor (<i>Oryza barthii</i>)	OGE/OMAP (NSF award #1026200)
<i>Oryza sativa ssp. japonica</i>	3,332,525	160K SNPs×20 accessions 1311 SNPs×395 accessions ~4 M BGI Japonica vs. Indica SNPs	McNally et al. [34] Zhao et al. [35] Yu et al. [36] NCBI dbSNP OGE/OMAP (NSF award #1026200)
<i>Oryza sativa ssp. indica</i>	4,747,883	~4 M BGI Japonica vs Indica SNPs	Yu et al. [36] NCBI dbSNP OGE/OMAP (NSF award #1026200)
<i>Sorghum bicolor</i>	257,153 SV: 64,507	265K SNPs×336 SAP lines Structural variants from Database of Genomic Variants archive (dGVA)	Morris et al. [37] Zheng et al. [38]
<i>Vitis vinifera</i>	457,404	Resequencing USDA germplasm collection	Myles et al. [31]
<i>Zea mays</i>	50,719,843	HapMap1: NAM founder lines HapMap2: 103 pre-domesticated & domesticated lines	Gore et al. [39] Chia et al. [40]

platform [47] supports the implementation of pathway databases in the BioCyc collection [48] to which Gramene has contributed MaizeCyc [45], RiceCyc [44], SorghumCyc [46], and BrachyCyc [46]. In addition, this resource mirrors six databases for *Arabidopsis* (AraCyc [49]), medicago (MedicCyc [50]), poplar (PoplarCyc [51]), potato (PotatoCyc [52]), coffee (CoffeaCyc [52]), and tomato (Lycocyc [52]), as well as the MetaCyc [48] and PlantCyc [51] reference databases (Fig. 3). The Plant Reactome is based on the Reactome data model and visualization platform [53]. It currently hosts manually curated rice and *Arabidopsis* pathways, and gene homology-driven inferred pathway projections for the maize and *Arabidopsis thaliana* reference genomes. It will continue to grow with the addition of data for new species and broader coverage of molecular interactions.

Search: All species for Go

e.g. Carboxy* or chx28

Popular genomes

- Arabidopsis thaliana** TAIR10
- Oryza sativa Japonica (Rice)** IRGSP-1.0
- Triticum aestivum** IWGSP1
- Hordeum vulgare** 030312v2
- Zea mays** AGP-v3
- Physcomitrella patens** ASM242v1

All genomes

-- Select a species --

[View full list of all Ensembl Plants species](#)

What's New in Release 40

- Updated genomes
 - Updated gene models for *Oryza sativa* from rap-db.
- New data
 - Gene models for *Triticum aestivum* (bread wheat) chromosome survey sequences from MIPS.
 - DNA-DNA alignments between bread wheat and *Oryza sativa*, as well as *Brachypodium distachyon*.
 - DNA-DNA alignments between *Hordeum vulgare* (barley) and the wild wheat progenitors (*Triticum urartu* and *Aegilops tauschii*).

Did you know...?

The **Assembly Converter Tool** allows the conversion of a variety of annotation formats between different versions of a genome assembly.

Bread wheat genome and gene annotation released!

The bread wheat genome in Ensembl Plants is the chromosome survey sequence for *Triticum aestivum* cv. Chinese Spring generated by the International Wheat Genome Sequencing Consortium. The gene models are provided by MIPS (version 2.0). A total of 108,569 protein coding genes have been predicted.

See also the wheat homepage at [Z. UGI](#)

[Read more about the assembly, annotation and analysis of bread wheat provided by Ensembl Plants...](#)

Organelle Annotation

For annotations relating to Organelles, see the [organelles page](#)

Fig. 2 Gramene Ensembl Genome Browser homepage (Ensembl software v75)

The Genomes and Pathway modules enable species-specific and cross-species data downloads for discrete region(s), gene(s) or gene feature(s) via the Genome Browser, and pathway-centered downloads via the Pathways portal and Plant Reactome. In addition, project data is available for customizable downloads from the GrameneMart utility (<http://ensembl.gramene.org/Tools/Blast?db=core> [54]), nucleotide and protein sequence alignments via BLAST (<http://ensembl.gramene.org/Tools/Blast?db=core>), bulk downloads via file transfer protocol (FTP) at Gramene (<ftp://ftp.gramene.org/pub/gramene> and Ensembl Genomes (<http://ensembl.gramene.org/info/web-site/ftp/index.html>), and programmatic access via Ensembl's REST application programming interface (API) and public MySQL (<http://www.gramene.org/web-services> [55]). Since March 2013, the website, database, and its contents are being updated quarterly and updates can be followed from the Gramene news portal (<http://www.gramene.org/blog>) and by browsing the site's release notes (<http://www.gramene.org/release-notes>).

This chapter summarizes updates to the Gramene website and database since reported in the last edition of this volume [43].

Gramene: Plant Biochemical Pathway Database and Home to RiceCyc, MaizeCyc, BrachyCyc and SorghumCyc.

Gramene: Plant Biochemical Pat...

pathway.gramene.org

Google

Gramene

Pathways Home | Search Pathways | Omics Viewer | Omics Validator | Downloads | Help | Tutorial | FAQs | Release notes | Plant Reactome (beta)

Plant Metabolic Pathways

The pathways section in the Gramene databases is home for RiceCyc, MaizeCyc, BrachyCyc and SorghumCyc, the pathway databases for rice, maize, *Brachypodium*, and sorghum, respectively. It also provides mirrors of pathway databases from *Arabidopsis*, tomato, potato, pepper, coffee, *Medicago*, *E. coli*, and the MetaCyc and PlantCyc reference databases, and might enable comparative analysis again upon software upgrade that supports the most current version of ptools software 17.0 by the original sources, such as the [SolGenomics Network](#) (SGN). In addition to search and browse functions, the database allows users to find genes mapped to respective reactions and pathways and draw inetspecific comparison between the pathways.

Pathways Browse and Other Options

Click on the species specific links such as **browse** to go through the list of pathways; **summary** to get a summarized overview. Click on the **more info** link to learn more details on the respective pathway database.

<p>RiceCyc ver 3.3 <i>Oryza sativa japonica</i> Strain: Nipponbare Browse Summary More Info</p>	<p>AraCyc* ver 10.0 <i>Arabidopsis thaliana</i> Strain: Columbia Browse Summary More Info</p>	<p>EcoCyc* ver 17.0 <i>Escherichia coli</i> Strain: K-12 MG1655 Browse Summary More Info</p>
<p>SorghumCyc ver 1.1 <i>Sorghum bicolor</i> Strain: BTx623 Browse Summary More Info</p>	<p>MedicCyc* ver 1.0.1 <i>Medicago truncatula</i>, Barrelclover Unavailable Browse Summary More Info</p>	<p>MetaCyc* ver 17.0 Reference Pathway Database Strain: not applicable Browse Summary More Info</p>
<p>MaizeCyc ver 2.2 <i>Zea mays</i> Strain: B73 Browse Summary More Info</p>	<p>PoplarCyc* ver 5.0 <i>Populus trichocarpa</i> (and other <i>Populus</i> species and hybrids) Strain: n/a Browse Summary More Info</p>	<p>PlantCyc* ver 7.0 Plant Metabolic Pathway Database Strain: not applicable Browse Summary More Info</p>
<p>BrachyCyc ver 2.0 <i>Brachypodium distachyon</i> Strain: Bd21 Browse Summary More Info</p>	<p>PotatoCyc* ver 1.0.1 <i>Solanum tuberosum</i>, Potato Strain: n/a Browse Summary More Info</p>	
	<p>CoffeaCyc* ver 1.1.1 <i>Coffea canephora</i>, Coffee Strain: n/a Browse Summary More Info</p>	

Fig. 3 Gramene's BioCyc pathways homepage. <http://pathway.gramene.org>

2 Materials

2.1 Hardware and Software Requirements for Users

A computer with internet access and a standard web browser such as Mozilla/Firefox, Internet Explorer, Chrome, or Safari.

2.2 Gramene System Components

Gramene is a web-based application that allows users to search and view biological data, making use where appropriate of graphics viewers such as the Ensembl genome browser or the Pathway Tools Omics viewer. Data is maintained in distinct relational databases (MySQL), and users connect to the site using a standard web browser. User queries for static (HTML) and dynamic content are negotiated by the Apache web server via the Solr search platform and a middleware layer written in Perl. Bulk downloads of data are provided through FTP sites at Gramene and Ensembl Genomes.

The image shows a screenshot of the Plant Reactome homepage. The browser window title is 'Reactome' and the address bar contains 'plantreactome.gramene.org/'. The page has a green and white color scheme. At the top, there is a navigation menu with links for 'Home', 'About', 'Documentation', 'Tools', and 'Contact Us'. Below this is a search bar with a magnifying glass icon and a list of search examples: 'LOC_Os01g45760.1', 'YUC4', 'cytokinins', and 'glucose'. There are buttons for 'Advanced search', 'Browse Pathways', and 'Retrieve SBML'. The main content area is divided into several sections. The 'About Plant Reactome' section describes the database as a freely accessible plant pathway database. The 'Featured pathway: Ethylene biosynthesis and signaling' section includes a complex metabolic pathway diagram. The 'Explore...' section lists various metabolic and regulatory pathways in *Oryza sativa*. The 'News and Notes' section contains several bullet points with links to new rice pathways, the 40th database release, a presentation at PAG XXII, a video tutorial, and useful links to other databases like KEGG, WikiPathways, MetaCyc, and Plant Metabolic Network.

Fig. 4 Gramene's Plant Reactome homepage. <http://plantreactome.gramene.org>

2.3 Local Installation of Gramene

The minimum hardware configuration required for a local installation of Gramene consists of a desktop or server with a multicore CPU, 4GB of memory, and 500GB of disk space. Installation inside a virtual machine is possible. A recent distribution of Linux is required, such as Redhat/CentOS 6 or Ubuntu 12.x. Software packages required include Apache web server (*see Note 1*), Perl, PHP, MySQL, OpenJava, Drupal, and Apache Solr. Many of these can be installed via the distribution package management system

(yum, apt-get). Solr can be downloaded from the Apache Solr website. For specific installation instructions, contact Gramene developers at feedback@gramene.org.

3 Methods

3.1 Basic Navigation of the Gramene Website

Gramene is powered by multiple modular platforms. The main entry point for the system is through the front web homepage (<http://www.gramene.org>; Fig. 1). Every Gramene page contains the main navigation bar and module-specific navigation bars if applicable, a simple search form that can be refined to interrogate the different data modules, a link to the homepage and another to our Contact page. The main navigation bars are found at the top and left side of each Gramene content page and constitute the main entry point to the search module (Search), genome browser (Genomes), pathway databases (Pathways), bulk data sets (Download), information about the project (About) and its collaborators (Collaborators), outreach events and educational materials (Outreach), and legacy data and resources (Archive). The Contact link is set up to provide the user a feedback page where the URL from the page that the user was viewing at the time of the response is automatically included in the message. The interfaces within Gramene are interactive, providing the user with links to external reference databases as well as links to internal modules within Gramene.

3.2 Example Uses of Gramene

Within the constraints of this chapter, it would not be possible to go through all of the Gramene interfaces. Instead, these examples provide sample queries and walk through using Gramene to obtain information and facilitate genomic research. These and additional examples focused on comparative analysis of plant metabolic and regulatory pathways as well as plant gene expression analysis are available from Gramene's Outreach page (www.gramene.org/outreach). Herein we will only demonstrate one of many possible ways to explore the Gramene website to address a given query and encourage users to discover other ingenious ways to solve them.

Exercise 1. View a phylogenetic tree for a family of transcription factors

In this exercise, we will navigate a phylogenetic tree for plant genes in the TCP family of transcription factors centered on the maize gene *tb1* (teosinte branched 1 [56]). We will then generate a list of homologues (i.e., orthologues and paralogues) for this gene, highlight species-specific homologues with particular Gene Ontology (GO) annotations in the tree, and download images and tables with the results. On the Gramene homepage, type “tb1” in the search box on the top of the page. Once the results appear, narrow your target by specifying “*Zea mays*” under Species. The resulting link will conduct you to the maize *tb1* gene page of the Ensembl genome

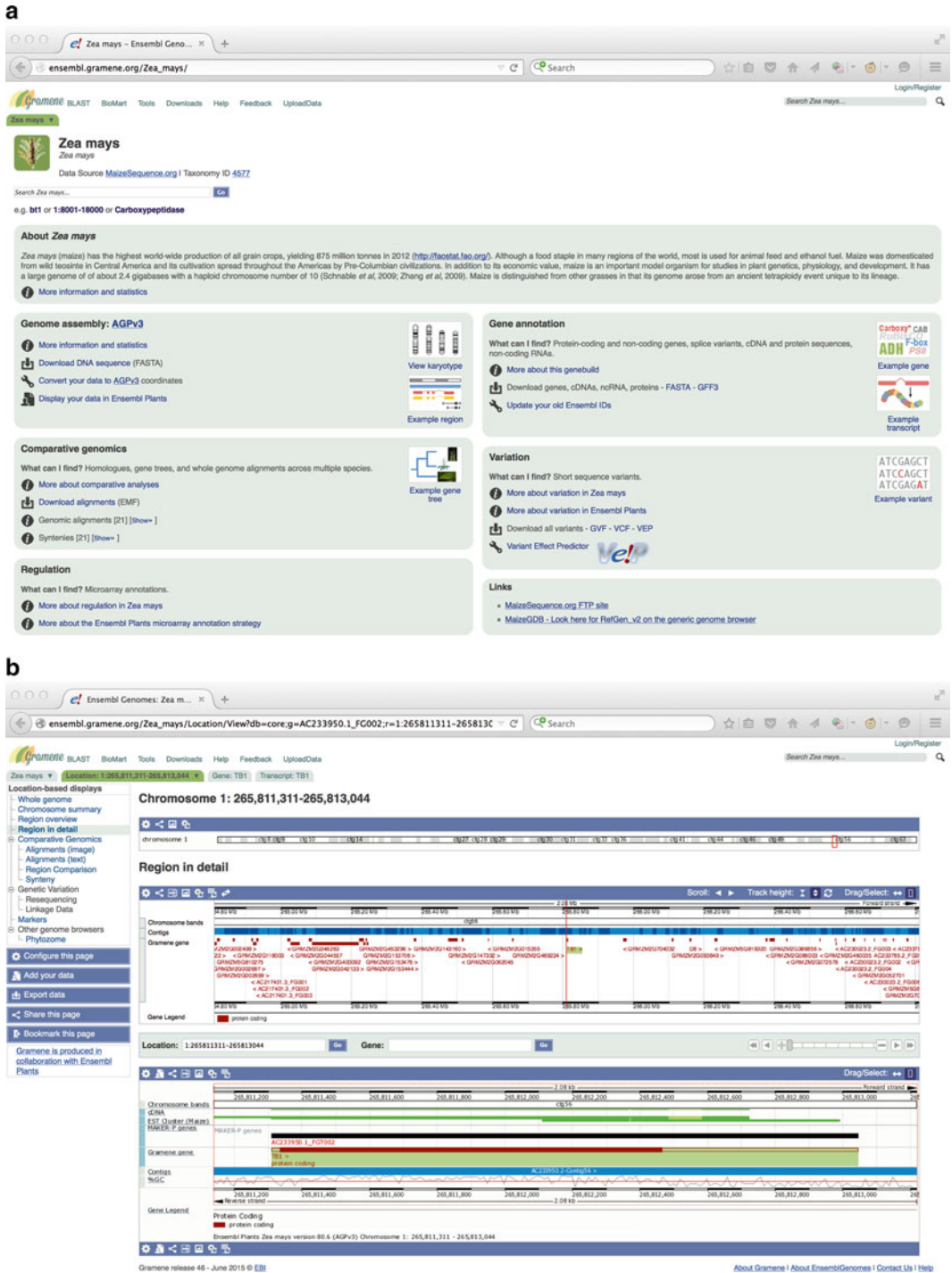


Fig. 5 Gramene Ensembl genome browser pages: (a) Species and (b) Location (e.g., maize *tb1* gene)

a

Ensembl Genomes: Zea m...
ensembl.gramene.org/Zea_mays/Gene/Summary?db=core&g=AC233950.1_FG002&v=1-265811311-265813

Gene: **TBI1** AC233950.1_FG002

Description: Transcription factor TEGOSANTE BRANCHED 1 [Source:UniProtKB/Swiss-Prot;Acc:Q89W02]

Location: Chromosome 1, 265,811,311-265,813,044 forward strand.

About this gene: This gene has 1 transcript (nucleotide), 87 orthologues and 17 paralogues.

Transcripts: [Hide transcript table](#)

Name	Transcript ID	bp	Protein	Biotype	UniProt	Flags
TBI1	AC233950.1_FG002	1754	276aa	Protein coding	A0A06CXX00 Q89W02	

Summary

Name: TBI1 (UniProtKB Gene Name)

UniProtKB: This gene has proteins that correspond to the following UniProtKB identifiers: [Q89W02](#)

Gene type: Protein coding

Annotation Method: Gene annotation by Gramene through an automated, evidence-based method

Go to Region in Detail for more tracks and navigation options (e.g. zooming)

Genome area: [Genomic track showing exons and introns]

Genes: [TBI1](#)

Transcripts: [TBI1](#)

Variation Legend: [Protein variant](#) [Synonymous variant](#) [3' prime UTR variant](#) [Upstream gene variant](#) [Downstream gene variant](#) [Intergenic variant](#)

Gene history: [Protein Coding](#) [Gene model](#)

Configuring the display: Tip: use the "Configure this page" link on the left to show additional data in this region.

Gramene release 48 - June 2019 © EBI

About Gramene | About EnsemblGenomes | Contact Us | Help

b

Ensembl Genomes: Zea m...
ensembl.gramene.org/Zea_mays/Transcript/SupportingEvidence?db=core&g=AC233950.1_FG002&v=1-265811311-265813

Transcript: **TBI1** AC233950.1_FG002

Description: Transcription factor TEGOSANTE BRANCHED 1 [Source:UniProtKB/Swiss-Prot;Acc:Q89W02]

Location: Chromosome 1, 265,811,311-265,813,044 forward strand.

About this transcript: This transcript has 1 exon, is annotated with 15 domains and features, is associated with 19 variations and maps to 1 oligo probe.

Gene: This transcript is a product of gene [AC233950.1_FG002](#) [Hide transcript table](#)

Supporting evidence

Genome area: [Genomic track showing exons and introns]

Transcript support: [TBI1](#)

Domains: [TBI1](#)

External data: [TBI1](#)

Personal annotation: [TBI1](#)

Gene history: [TBI1](#)

Protein history: [TBI1](#)

Configuring the page: Tip: use the "Configure this page" link on the left to show additional data in this region.

Gramene release 48 - June 2019 © EBI

About Gramene | About EnsemblGenomes | Contact Us | Help

Fig. 6 Gene-centric Gramene Ensembl genome browser pages: (a) Gene, (b) Transcript, and (c) Variation pages (e.g., maize *tbi1* gene; SNP variant PZE01264848659)

c

Ensembl Genomes: Zea mays x

ensembl.gramene.org/Zea_mays/Variation/Explore?db=core;g=AC233950.1_FG002;r=1:265811311-265813044;t=AC233950.1_FG... Login/Register

Gramene BLAST BioMart Tools Downloads Help Feedback UploadData Search Zea mays...

Zea mays Location: 1:265,811,311-265,813,044 Gene: TB1 Transcript: TB1 Variation: PZE01264848659

Variation displays

- Explore this variation
- Genomic context
 - Genes and regulation
 - Flanking sequence
- Genotype frequency
- Individual genotypes
- Linkage disequilibrium
- Phenotype Data
- Phylogenetic Context
- Citations
- External Data

Configure this page

Add your data

Export data

Share this page

Bookmark this page

Gramene is produced in collaboration with Ensembl Plants

PZE01264848659 SNP

Original source Alleles **G/A** | Ambiguity code: R

Location Chromosome 1:265811476 (forward strand) | [View in location tab](#)

Most severe consequence **Missense variant** | [See all predicted consequences \(Genes and regulation\)](#)

HGVS names This variation has 3 HGVS names - click the plus to show

About this variant This variant overlaps 1 [transcript](#).

Explore this variation

Genomic context

Genes and regulation

Genotype frequency

Individual genotypes

Linkage disequilibrium

Phenotype data

Citations

Phylogenetic context

Flanking sequence

Fig. 6 (continued)

browser. Note the four distinct tabs on the top of this page: Species (Fig. 5a), Location (Fig. 5b), Gene (Fig. 6a), and Transcript (Fig. 6b); an additional Variation page (Fig. 6c) may be accessed for species with variation data in Gramene (*see* Table 2). Each of these tab views will be discussed in more detail below. Common to the Location, Gene, Transcript, and Variation pages, as well as the views available therein, are customizable tracks, links to internal pages, and contextual links to data sources outside of Gramene. Actions enabled for each of those pages and their embedded views include (1) configuring and resizing, (2) uploading and managing user-provided data for graphic display, (3) exporting or downloading data, and (4) sharing pages and images. For example, you may customize the tracks on display by selecting the “Configure this page” instruction on the left side navigation bar or upon clicking on the “Configure this image” icon on the top left corner of an image. A new browser window will pop up listing all available data tracks for the browser view that you wish to customize. Data tracks are grouped by category; click on a category to see the complete list of available tracks for that category (e.g., “mRNA and protein alignments” may include tracks for EST clusters, cDNAs, and protein features from various species, sources, or methods). A track gets activated for display on the browser by clicking on the square preceding its name

and selecting a desired “track style”. Favorite tracks may be set and the order of tracks may be changed. Save your selections and close the pop-up window by clicking on the check mark on the top right corner. The browser will automatically refresh itself and your selected tracks should now be visible.

Gramene Ensembl Genome Browser pages:

1. The Species page (*Zea mays* for this example; Fig. 5a) contains detailed information about the reference genome assembly and gene annotation; comparative genomics data including phylogenetic gene trees, whole-genome alignments, and synteny views; gene regulation (microarray) data; genetic and structural variation; and links to download data sets in bulk.
2. The Location page (Chr 1: 265,811,311–265,813,044 in the B73 maize AGPv3 assembly; Fig. 5b) offers several scalable views on the left side navigation bar, e.g., karyotype or whole-genome view, chromosome summary, region overview, region in detail (expanded red box from the region overview), as well as comparative genomics views, which include multi-species alignments, region comparisons, and synteny views. Semantic zooming is available for each “region” view.
3. The Gene page (TBI; Fig. 6a) provides a summary of data available for a given gene, as well as an extensive list of features including splice variants (*see* also Transcript page), exon/intron marked-up sequence, associated ontology terms and literature references, external references, comparative genomic alignments, expandable gene trees, orthologues and paralogues, and genetic/structural variation. The Plant Compara Gene Trees are derived from a pre-computed phylogenetic analysis of protein-coding genes from all Gramene species, plus several representative animal genomes used as outgroups. The Pan-taxonomic Compara Gene Trees sample species more broadly across taxa represented by the Ensembl Genomes project, including bacteria, fungi, protists, and metazoa, and include only a subset of representative plant species held within the Gramene database.
4. The Transcript page (TBI-201; Fig. 6b) includes sequence data, external cross-references (including oligo probe sources), supporting protein/EST evidence, GO associations, variation, and protein domains and features. If variation data is available for a given gene, each variant will have its own Variation page (*see* below). The Variations table under the Protein Information category provides a complete list of the transcript’s variants with alleles, functional consequence, relative position in the protein’s amino acid sequence, ambiguity code, and actual affected codons/amino acids, if any. Moreover, for species like *Arabidopsis* in which the same set of variants have been genotyped in different populations, tabular and graphic “population comparisons” are available from the Transcript page.

5. The Variation page (e.g., PZE01264848659 for *tb1*; Fig. 6c) includes the variant's genomic context, functional consequences in all transcripts, individual genotype data, as well as allele/genotype frequency by population tables. Note that if several transcripts are available for a given gene, the same variant may have different functional consequences in each transcript as per its relative location in the corresponding protein product.

To view the phylogenetic tree for the TCP family of transcription factors, go to the *tb1* Gene page and click on the “Gene tree (image)” view. In the “Highlight annotations” table, both InterPro and GO terms are enabled by default; uncheck the box for GO terms to show only InterPro domains. From the list, select IPR005333, which is the InterPro ID for the complete TCP domain (*see Note 2* for visualizing the complete protein domain structure of the maize *tb1* gene). Figure 7a displays the collapsed view of the tree with all the clades highlighted. Click on “View fully expanded tree” from the “View options” at the bottom of the page. Except for a handful of genes, all the genes in the tree image will light up because of the prevalence of the TCP domain. We may also highlight orthologues and paralogues between two species. For example, let's find the sorghum orthologue with highest similarity to maize *tb1*. From the maize gene's page, select “Plant Comparison Orthologues” and enter “sorghum” in the “Filter” box on the top right corner of the orthologues table (Fig. 7b). In the “Compare” column, click on the “Gene tree (image)” link for the sorghum orthologous gene (Sb01g010690), and upon full expansion of the tree, you will see TB1 and SB01G010690 highlighted in different shades of red, maize within-species paralogues in different shades of blue, and sorghum paralogues highlighted in black (Fig. 7c shows the collapsed view of the highlighted tree). Note that by clicking on any speciation tree node, a pop-up inset will appear with various parameters describing the tree, as well as options to selectively collapse nodes and view a sub-tree in other formats like FASTA.

Exercise 2. Explore genetic variation in the rice orthologues of a maize gene with a known trait association

We will now explore genetic variation in the rice orthologues of the maize lycopene epsilon gene (*lcyE*). Specifically, we will determine whether the non-synonymous substitution mapping to nucleotide 210 relative to the start codon of the transcript with the longest genomic span (LCYE-201 or GRMZM2G012966_T03), which was found to be associated with provitamin A accumulation in the maize kernel [57], is also present in its rice orthologues.

Go to the maize *lcyE* gene page as done for the *tb1* gene. From the gene and transcript pages, you may visualize all its genetic variants in tabular (“Variation table” option in the left side navigation bar; Fig. 8a) or graphic form in their genomic context

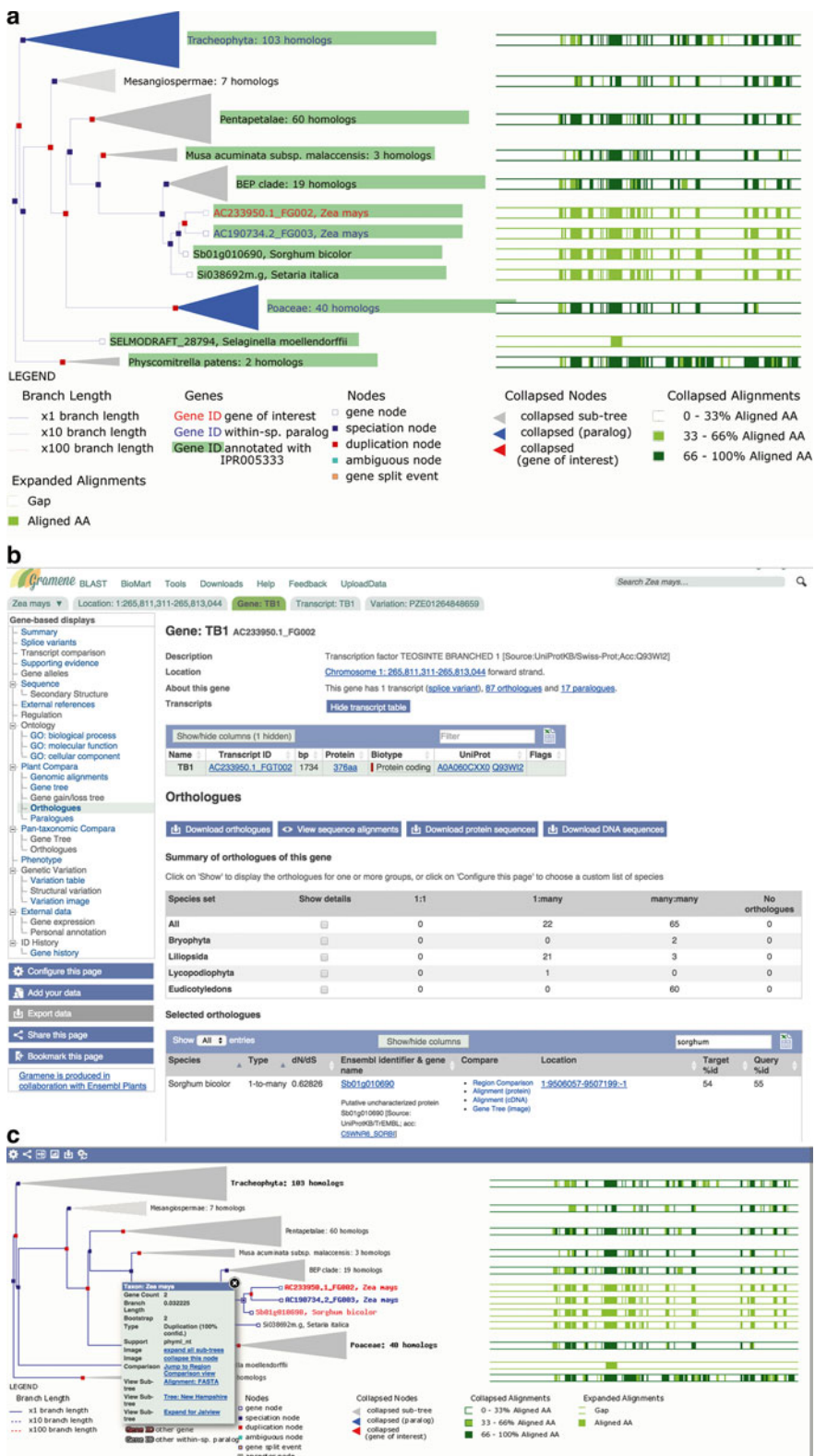


Fig. 7 Exercise 1: visualization of a phylogenetic tree for the TCP family of transcription factors centered on maize TB1. **(a)** Collapsed view of the tree highlighting all gene products that include the TCP InterPro domain (IPR005333). **(b)** Filtered view of sorghum orthologues of TB1. **(c)** Gene tree image for TB1 highlighting its sorghum orthologues (e.g., Sb01G010690) and within-species paralogues (maize and sorghum, respectively); speciation nodes shown in black, duplication nodes shown in red. Also shown is *inset* that pops up upon clicking on any node

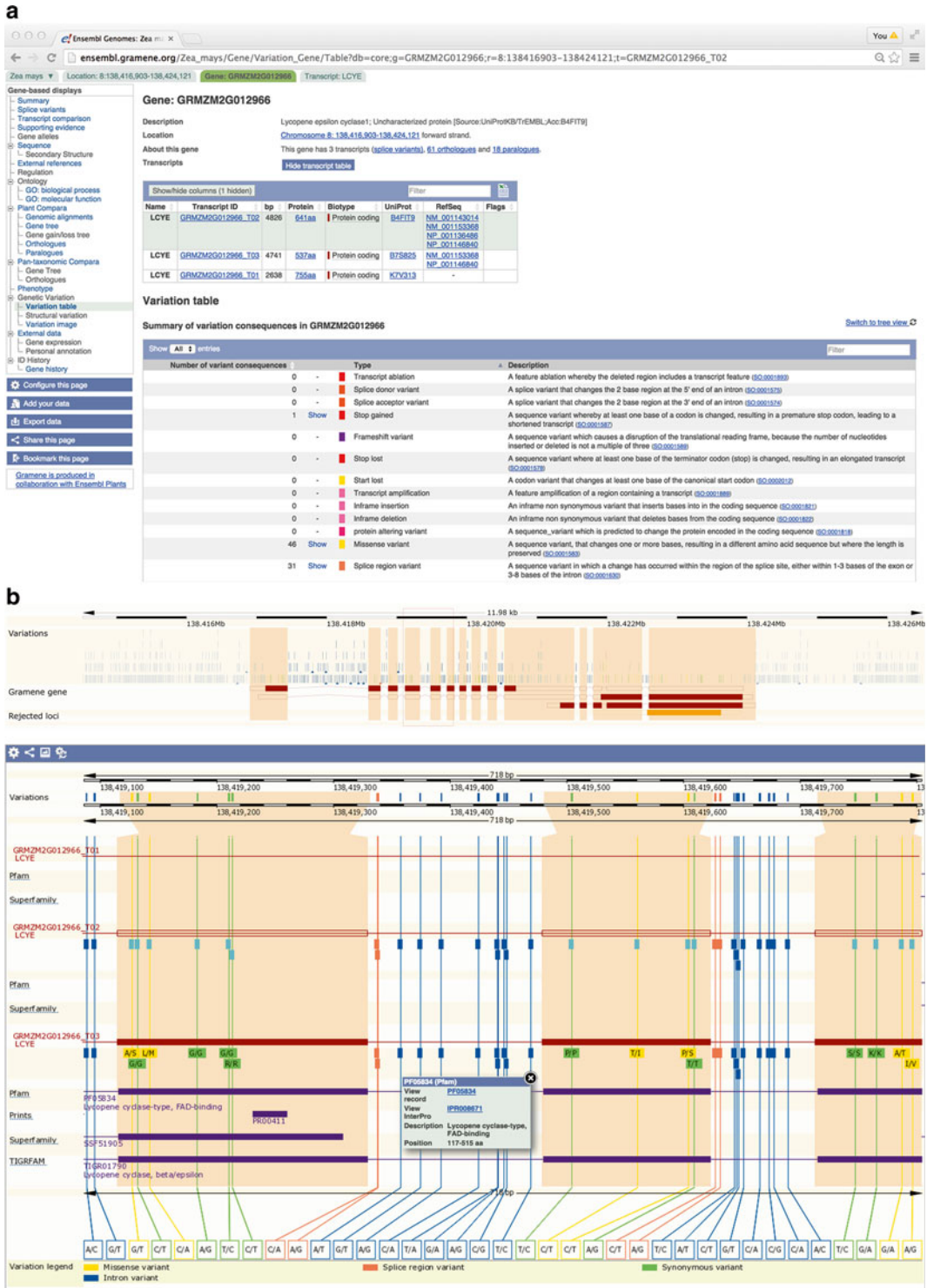


Fig. 8 Exercise 2: exploring genetic variation in a rice gene while looking for conservation of a maize SNP variant (PZE08137569063) associated with provitamin A accumulation in the kernel [57]. (a) Genetic variation in the maize *lycE* gene in tabular form, and (b) graphic form

(“Variation image” option; Fig. 8b). The Variation table groups variants by functional consequence; by clicking on the “Show” option for a given category (e.g., missense variant), you will get a list of the variants with other data like genomic position, alleles, relative amino acid position (if affected), etc. The Variation image displays the same information as the table in graphic form, plus the relative location of known protein domains. However, if you know the SNP identifier, the simplest way to find all the available information for a given variant is to select the “Variations” option under “Protein Information” as it lists all variants by identifier. The SNP variant associated with provitamin A accumulation identified by Harjes *et al* [57] is PZE08137569063. As shown in the “Genotype frequency” (Fig. 9a) view available from the Variation page, this variant has alleles G and T with variable genotype frequencies in 13 maize or teosinte populations, including HapMap2 (“Zmays”).

Now, let’s identify the closest rice orthologues of the *lcyE* gene by proceeding as described above in the “Gene tree (image)”. The “Orthologues” view allows users to download all or a selected set of orthologous genes (by using the filter box on the top right corner of the table), as well as to view and download the corresponding protein sequences and/or pairwise alignments. To download nucleotide sequences or download all the genetic variants for the orthologues, users could go to each individual gene’s page and proceed as described above (i.e., go to the Variation table/image and download the data directly from the table/image or click on “Export data” option on the left sidebar menu). Alternatively, users may download for each species the same DNA/protein sequence and variation data using, respectively, the “Plant Genes” and “Plant Variation” databases in the GrameneMart utility (<http://ensembl.gramene.org/biomart/mart-view/>; Fig. 9b). Users may visually compare all the species in the gene tree by selecting “Gene tree (alignment)” in the left sidebar menu or view pairwise genomic alignments with the “Genomic alignments (text)” option. Alternatively, users may use a multiple alignment program such as ClustalW to visually compare the rice orthologous gene sequences, and realize that (1) this site has not been found to be polymorphic in *O. sativa Japonica* and *Indica*, (2) there is a sequence gap around this position in *O. glaberrima*, and (3) the ancestral G allele is the one present at this position in the *O. nivara*, *O. glumaepatula*, and *O. punctata* orthologous genes.

Further genomic analysis may be performed with the Ensembl “Tools” available at <http://ensembl.gramene.org/tools.html> and other links from Gramene’s archival Diversity pages at <http://archive.gramene.org/diversity/tools.html>.

Exercise 3. Upload, visualize, and share your own data into a new genome browser track

The Ensembl genome browser allows users to upload their own data and visualize it on a custom track. Data may be formatted in various file formats including GFF, GTF, BED, BAM, VCF,

a

Zea mays Location: 8:138,416,835-138,417,835 Variation: PZE08137569063

Variation displays

- Explore this variation
- Genomic context
- Genes and regulation
- Flanking sequence
- Genotype frequency**
- Individual genotypes
- Linkage disequilibrium
- Phenotype Data
- Phylogenetic Context
- Citations
- External Data

PZE08137569063 SNP

Original source Alleles T/G | Ambiguity code: K

Location Chromosome 8:138417335 (forward strand) | [View in location tab](#)

Most severe consequence Missense variant | [See all predicted consequences \(Genes and regulation\)](#)

Synonyms Panzea_2.7GBS S8_138883026

HGVS names This variation has 4 HGVS names - click the plus to show

About this variant This variant overlaps 3 transcripts.

Genotype frequency

Frequency data (13)

Population	Allele: frequency (count)		Genotype: frequency (count)		
BREAD	T: 0.041 (450)	G: 0.959 (10642)	TIT: 0.019 (105)	GIG: 0.938 (5201)	GIT: 0.043 (240)
IBM	T: 0.351 (253)	G: 0.649 (467)	TIT: 0.339 (122)	GIG: 0.636 (229)	GIT: 0.025 (9)
IBM_parent	T: 0.351 (253)	G: 0.649 (467)	TIT: 0.339 (122)	GIG: 0.636 (229)	GIT: 0.025 (9)
Margaret_Smith_lines	T: 0.594 (253)	G: 0.406 (173)	TIT: 0.563 (120)	GIG: 0.376 (80)	GIT: 0.061 (13)
NAM	T: 0.706 (8104)	G: 0.294 (3380)	TIT: 0.690 (3962)	GIG: 0.279 (1600)	GIT: 0.031 (180)
NAM_F1	T: 0.630 (29)	G: 0.370 (17)	TIT: 0.304 (7)	GIG: 0.043 (1)	GIT: 0.652 (15)
NAM_common_parent	T: 1.000 (110)	G: 0.000 (0)	TIT: 1.000 (55)		
NAM_parent	T: 0.441 (82)	G: 0.559 (104)	TIT: 0.441 (41)	GIG: 0.559 (52)	
Spanish_inbred_lines	T: 0.421 (69)	G: 0.579 (95)	TIT: 0.341 (28)	GIG: 0.500 (41)	GIT: 0.159 (13)
W22_x_Teosinte_BC2S3	T: 1.000 (182)	G: 0.000 (0)	TIT: 1.000 (91)		
Zmays	T: 0.262 (44)	G: 0.738 (124)	TIT: 0.250 (21)	GIG: 0.726 (61)	GIT: 0.024 (2)
fine_mapping	T: 1.000 (2)	G: 0.000 (0)	TIT: 1.000 (1)		
teosinte_inbred_lines	T: 0.406 (13)	G: 0.594 (19)	TIT: 0.312 (5)	GIG: 0.500 (8)	GIT: 0.188 (3)

b

Gramene BLAST BioMart Tools Feedback Search all species...

New Count Results URL XML Perl Help

Dataset 43 / 5512746 SNPs
Oryza sativa Japonica (Rice)
 variations (IRGSP-1.0 (IRGSP-1.0))

Filters

Ensembl Gene ID(s) [Max 500 advised] : [ID-list specified]

Attributes

- Variation Name
- Chromosome name
- Position on Chromosome (bp)
- Strand
- Variant Allele
- Population Name
- Population Size
- Population Genotype
- Genotype Frequency
- Ensembl Gene ID
- Ensembl Transcript ID
- Consequence to transcript

Dataset
 [None Selected]

Please restrict your query using criteria below

REGION:

GENERAL VARIATION FILTERS:

GENE ASSOCIATED VARIATION FILTERS:

Ensembl Gene ID(s) [Max 500 advised] OS01G0581400

No file selected.

Consequence type

- splice_acceptor_variant
- splice_donor_variant
- stop_lost
- coding_sequence_variant
- missense_variant

Fig. 9 (a) Genotype frequency data for PZE08137569063 in 13 maize or teosinte populations from HapMap2 and the Panzea 2.7 GBS data release Gramene. **(b)** Genotype data for Gene sequences, orthologous/paralogous gene lists, and gene variants available for customized download via the GrameneMart

bedGraph, gbrowse, PSL, WIG, BigBed, BigWig, and TrackHub. Some data like GFF annotations may be directly uploaded from a local machine. Large data files like BED/BAM alignments or BigWig graphic display configurations need to be uploaded onto a local server that is accessible to the browser via a URL. Another way to share third-party data is via a DAS (Distributed Annotation System) registry, which would need to be set up by a software engineer.

Test data sets consisting of BAM alignments and CpG methylation for B73 and Mo17 maize lines used in the study by Regulski et al. [58] are available from the Gramene outreach pages to upload and visualize for this exercise. To upload the data simply click on the “Add/Manage your data” option on the left bar menu of any genome browser page (Fig. 10a). This action will take you to the upload page (Fig. 10b) where you need to specify the format of the file you intend to upload (formats and test sets are also described in the “Help on supported formats, display types, etc.” link therein). To visualize custom data in a new browser track, make sure that your track is turned on in the configuration menu and you are looking at a region that includes the new data you have just uploaded. The BAM alignments and CpG methylation ratios are shown in Fig. 10c.

4 Notes

1. Apache 2.x is not supported yet due to significant differences in the persistent Perl interpreter module (`mod_perl`).
2. For the complete protein domain structure of a gene, go to the Transcript page and select “Domains & features”. By clicking on a particular “Display all genes with this domain” link, you will get a list of all genes in the same species that contain any given InterPro domain. To download the list, click on the “Download” icon to the right of the Filter box.

Acknowledgements

The authors would like to thank all members of the Gramene Project, especially Bo Wang for going through the exercises and providing feedback for clarity in the protocols and Peter van Buren for system technology support. We are also grateful to Gramene’s users for valuable suggestions, and our collaborators for sharing genomic-scale data sets that make Gramene an outstanding community resource. The Genomes and Pathways modules in Gramene would not have been possible without the synergistic collaborations with the Ensembl Genomes project at the EMBL-European Bioinformatics Institute, and the Reactome project at the Ontario Institute for Cancer Research, respectively.

Gramene is supported by an NSF grant (IOS-1127112).

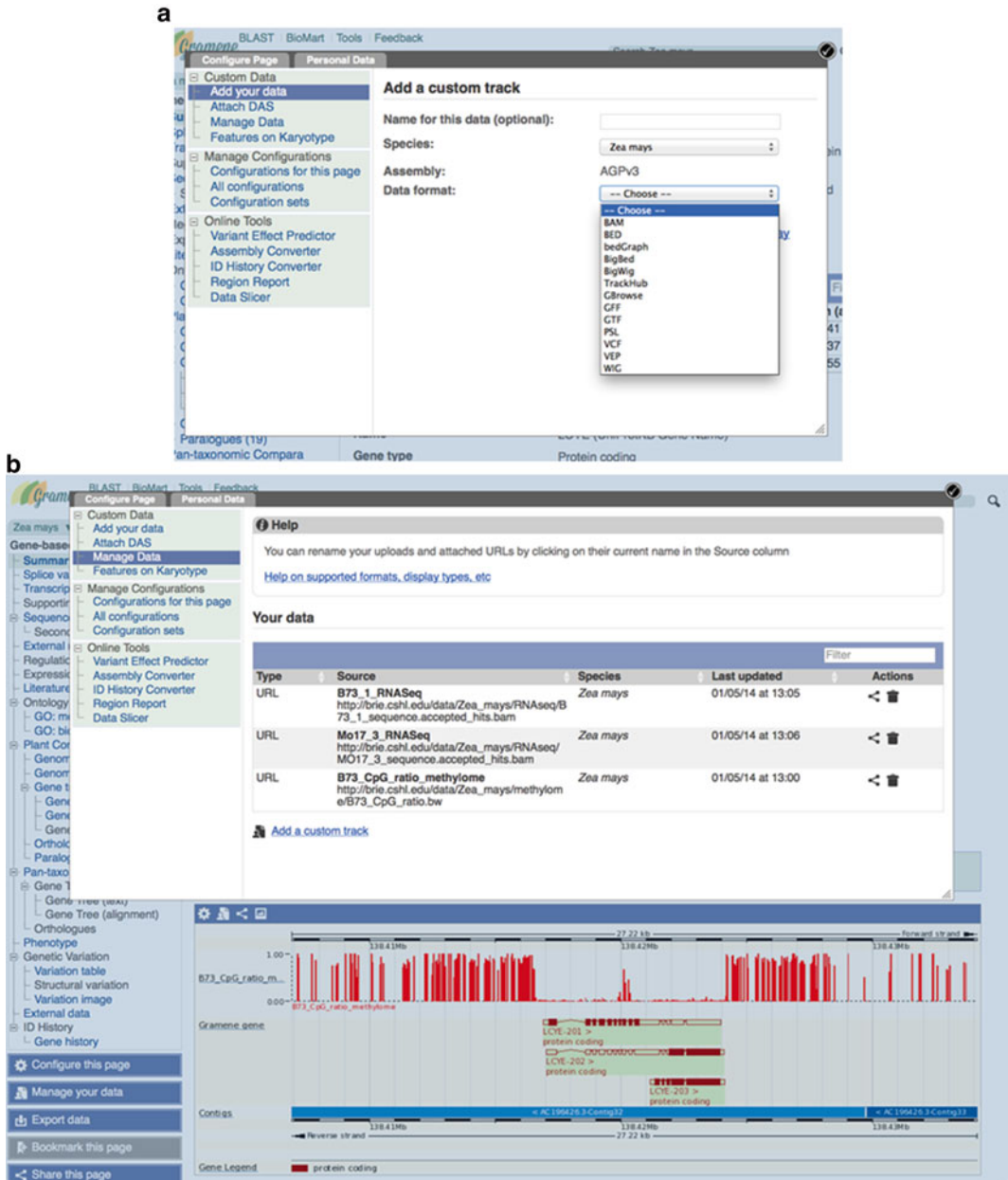


Fig. 10 Uploading and managing user-provided data to display as a new genome browser track. (a) Pop-up window upon clicking on the “Add/Upload Data” link from the Gene page. User selects species and file format for the data to be uploaded. (b) Preloaded BAM alignments and CpG methylation ratios. New track shows CpG methylation data in the selected gene region

References

1. Jia J, Zhao S, Kong X et al (2013) Aegilops tauschii draft genome sequence reveals a gene repertoire for wheat adaptation. Nature 496(7443):91–95
2. Amborella Genome Project (2013) The Amborella genome and the evolution of flowering plants. Science 342(6165): 1241089
3. Chamala S, Chanderbali AS, Der JP et al (2013) Assembly and validation of the genome

- of the nonmodel basal angiosperm *Amborella*. *Science* 342(6165):1516–1517
4. Hu TT, Pattyn P, Bakker EG et al (2011) The *Arabidopsis lyrata* genome sequence and the basis of rapid genome size change. *Nat Genet* 43(5):476–481
 5. The *Arabidopsis* Genome Initiative (2000) Analysis of the genome sequence of the flowering plant *Arabidopsis thaliana*. *Nature* 408(6814):796–815
 6. The International Brachypodium Initiative (2010) Genome sequencing and analysis of the model grass *Brachypodium distachyon*. *Nature* 463(7282):763–768
 7. Wang X, Wang H, Wang J et al (2011) The genome of the mesopolyploid crop species *Brassica rapa*. *Nat Genet* 43(10):1035–1039
 8. Merchant SS, Prochnik SE, Vallon O et al (2007) The *Chlamydomonas* genome reveals the evolution of key animal and plant functions. *Science* 318(5848):245–250
 9. Matsuzaki M, Misumi O, Shin IT et al (2004) Genome sequence of the ultrasmall unicellular red alga *Cyanidioschyzon merolae* 10D. *Nature* 428(6983):653–657
 10. Schmutz J, Cannon SB, Schlueter J et al (2010) Genome sequence of the palaeopolyploid soybean. *Nature* 463(7278):178–183
 11. International Barley Genome Sequencing Consortium, Mayer KF, Waugh R et al (2012) A physical, genetic and functional sequence assembly of the barley genome. *Nature* 491(7426):711–716
 12. Young ND, Debelle F, Oldroyd GE et al (2011) The *Medicago* genome provides insight into the evolution of rhizobial symbioses. *Nature* 480(7378):520–524
 13. D'Hont A, Denoeud F, Aury JM et al (2012) The banana (*Musa acuminata*) genome and the evolution of monocotyledonous plants. *Nature* 488(7410):213–217
 14. Chen J, Huang Q, Gao D et al (2013) Whole-genome sequencing of *Oryza brachyantha* reveals mechanisms underlying *Oryza* genome evolution. *Nat Commun* 4:1595
 15. Yu J, Hu S, Wang J et al (2002) A draft sequence of the rice genome (*Oryza sativa* L. ssp. *indica*). *Science* 296(5565):79–92
 16. Zhao W, Wang J, He X et al (2004) BGI-RIS: an integrated information resource and comparative analysis workbench for rice genomics. *Nucleic Acids Res* 32(Database issue):D377–D382
 17. International Rice Genome Sequencing Project (2005) The map-based sequence of the rice genome. *Nature* 436(7052):793–800
 18. Kawahara Y, de la Bastide M, Hamilton JP et al (2013) Improvement of the *Oryza sativa* Nipponbare reference genome using next generation sequence and optical map data. *Rice (N Y)* 6(1):4
 19. Rensing SA, Lang D, Zimmer AD et al (2008) The *Physcomitrella* genome reveals evolutionary insights into the conquest of land by plants. *Science* 319(5859):64–69
 20. Tuskan GA, Difazio S, Jansson S et al (2006) The genome of black cottonwood, *Populus trichocarpa* (Torr. & Gray). *Science* 313(5793):1596–1604
 21. The International Peach Genome Initiative, Verde I, Abbott AG et al (2013) The high-quality draft genome of peach (*Prunus persica*) identifies unique patterns of genetic diversity, domestication and genome evolution. *Nat Genet* 45(5):487–494
 22. Banks JA, Nishiyama T, Hasebe M et al (2011) The *Selaginella* genome identifies genetic changes associated with the evolution of vascular plants. *Science* 332(6032):960–963
 23. Bennetzen JL, Schmutz J, Wang H et al (2012) Reference genome sequence of the model plant *Setaria*. *Nat Biotechnol* 30(6):555–561
 24. Zhang G, Liu X, Quan Z et al (2012) Genome sequence of foxtail millet (*Setaria italica*) provides insights into grass evolution and biofuel potential. *Nat Biotechnol* 30(6):549–554
 25. Tomato Genome C (2012) The tomato genome sequence provides insights into fleshy fruit evolution. *Nature* 485(7400):635–641
 26. Potato Genome Sequencing Consortium, Xu X, Pan S et al (2011) Genome sequence and analysis of the tuber crop potato. *Nature* 475(7355):189–195
 27. Paterson AH, Bowers JE, Bruggmann R et al (2009) The *Sorghum bicolor* genome and the diversification of grasses. *Nature* 457(7229):551–556
 28. Brenchley R, Spannagl M, Pfeifer M et al (2012) Analysis of the bread wheat genome using whole-genome shotgun sequencing. *Nature* 491(7426):705–710
 29. Ling HQ, Zhao S, Liu D et al (2013) Draft genome of the wheat A-genome progenitor *Triticum urartu*. *Nature* 496(7443):87–90
 30. Jaillon O, Aury JM, Noel B et al (2007) The grapevine genome sequence suggests ancestral hexaploidization in major angiosperm phyla. *Nature* 449(7161):463–467
 31. Myles S, Chia JM, Hurwitz B et al (2010) Rapid genomic characterization of the genus *vitis*. *PLoS One* 5(1), e8219
 32. Atwell S, Huang YS, Vilhjalmsón BJ et al (2010) Genome-wide association study of 107 phenotypes in *Arabidopsis thaliana* inbred lines. *Nature* 465(7298):627–631

33. Fox SE, Preece J, Kimbrel JA et al (2013) Sequencing and de novo transcriptome assembly of *Brachypodium sylvaticum* (Poaceae). *Appl Plant Sci* 1(3):1200011
34. McNally KL, Childs KL, Bohnert R et al (2009) Genomewide SNP variation reveals relationships among landraces and modern varieties of rice. *Proc Natl Acad Sci U S A* 106(30):12273–12278
35. Zhao K, Wright M, Kimball J et al (2010) Genomic diversity and introgression in *O. sativa* reveal the impact of domestication and breeding on the rice genome. *PLoS One* 5(5): e10780
36. Yu J, Wang J, Lin W et al (2005) The genomes of *Oryza sativa*: a history of duplications. *PLoS Biol* 3(2), e38
37. Morris GP, Ramu P, Deshpande SP et al (2013) Population genomic and genome-wide association studies of agroclimatic traits in sorghum. *Proc Natl Acad Sci U S A* 110(2):453–458
38. Zheng LY, Guo XS, He B et al (2011) Genome-wide patterns of genetic variation in sweet and grain sorghum (*Sorghum bicolor*). *Genome Biol* 12(11):R114
39. Gore MA, Chia JM, Elshire RJ et al (2009) A first-generation haplotype map of maize. *Science* 326(5956):1115–1117
40. Chia JM, Song C, Bradbury PJ et al (2012) Maize HapMap2 identifies extant variation from a genome in flux. *Nat Genet* 44(7): 803–807
41. Flicek P, Amode MR, Barrell D et al (2014) Ensembl 2014. *Nucleic Acids Res* 42(Database issue):D749–D755
42. Kersey PJ, Allen JE, Christensen M et al (2014) Ensembl Genomes 2013: scaling up access to genome-wide data. *Nucleic Acids Res* 42(Database issue):D546–D552
43. Ware D (2007) Gramene. *Methods Mol Biol* 406:315–329
44. Dharmawardhana P, Ren L, Amarasinghe V et al (2013) A genome scale metabolic network for rice and accompanying analysis of tryptophan, auxin and serotonin biosynthesis regulation under biotic stress. *Rice (N Y)* 6(1):15
45. Monaco MK, Sen TZ, Dharmawardhana PD et al (2013) Maize metabolic network construction and transcriptome analysis. *Plant Genome* 6(1):1–12
46. Youens-Clark K, Buckler E, Casstevens T et al (2011) Gramene database in 2010: updates and extensions. *Nucleic Acids Res* 39(Database issue):D1085–D1094
47. Karp PD, Paley SM, Krummenacker M et al (2010) Pathway Tools version 13.0: integrated software for pathway/genome informatics and systems biology. *Brief Bioinform* 11(1):40–79
48. Caspi R, Altman T, Billington R et al (2014) The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases. *Nucleic Acids Res* 42(Database issue):D459–D471
49. Mueller LA, Zhang P, Rhee SY (2003) AraCyc: a biochemical pathway database for *Arabidopsis*. *Plant Physiol* 132(2):453–460
50. Urbanczyk-Wochniak E, Sumner LW (2007) MedicCyc: a biochemical pathway database for *Medicago truncatula*. *Bioinformatics* 23(11): 1418–1423
51. Zhang P, Dreher K, Karthikeyan A et al (2010) Creation of a genome-wide metabolic pathway database for *Populus trichocarpa* using a new approach for reconstruction and curation of metabolic pathways for plants. *Plant Physiol* 153(4):1479–1491
52. Bombarely A, Menda N, Teclé IY et al (2011) The Sol Genomics Network (solgenomics.net): growing tomatoes using Perl. *Nucleic Acids Res* 39(Database issue):D1149–D1155
53. Croft D, O’Kelly G, Wu G et al (2011) Reactome: a database of reactions, pathways and biological processes. *Nucleic Acids Res* 39(Database issue):D691–D697
54. Spooner W, Youens-Clark K, Staines D et al (2012) GrameneMart: the BioMart data portal for the Gramene project. *Database (Oxford)* 2012:bar056
55. Monaco MK, Stein J, Naithani S et al (2014) Gramene 2013: comparative plant genomics resources. *Nucleic Acids Res* 42(Database issue):D1193–D1199
56. Doebley J, Stec A, Hubbard L (1997) The evolution of apical dominance in maize. *Nature* 386(6624):485–488
57. Harjes CE, Rocheford TR, Bai L et al (2008) Natural genetic variation in lycopene epsilon cyclase tapped for maize biofortification. *Science* 319(5861):330–333
58. Regulski M, Lu Z, Kendall J et al (2013) The maize methylome influences mRNA splice sites and reveals widespread paramutation-like switches guided by small RNA. *Genome Res* 23(10):1651–1662

Chapter 8

PGSB/MIPS Plant Genome Information Resources and Concepts for the Analysis of Complex Grass Genomes

Manuel Spannagl, Kai Bader, Matthias Pfeifer, Thomas Nussbaumer, and Klaus F.X. Mayer

Abstract

PGSB (Plant Genome and Systems Biology; formerly MIPS—Munich Institute for Protein Sequences) has been involved in developing, implementing and maintaining plant genome databases for more than a decade. Genome databases and analysis resources have focused on individual genomes and aim to provide flexible and maintainable datasets for model plant genomes as a backbone against which experimental data, e.g., from high-throughput functional genomics, can be organized and analyzed. In addition, genomes from both model and crop plants form a scaffold for comparative genomics, assisted by specialized tools such as the CrowsNest viewer to explore conserved gene order (synteny) between related species on macro- and micro-levels.

The genomes of many economically important Triticeae plants such as wheat, barley, and rye present a great challenge for sequence assembly and bioinformatic analysis due to their enormous complexity and large genome size. Novel concepts and strategies have been developed to deal with these difficulties and have been applied to the genomes of wheat, barley, rye, and other cereals. This includes the GenomeZipper concept, reference-guided exome assembly, and “chromosome genomics” based on flow cytometry sorted chromosomes.

Key words PlantsDB, Wheat genome, Barley genome, GenomeZipper, CrowsNest synteny browser

1 Introduction

Within the first section of this chapter we describe peculiarities of Triticeae genomes and their consequences for genome assembly and analysis. Strategies and concepts to overcome analytical limitations associated with large genome size and complex genetics are outlined and results from the analysis of the bread wheat genome are highlighted.

In the second part of the chapter we introduce the content, technical setup, and architecture of the plant genome resources available at PGSB/MIPS. This includes the PGSB/MIPS PlantsDB

database platform as well as tools and views for the (comparative) analysis of plant genomes and transcriptome data.

2 Materials

2.1 Analyzing Complex Grass Genomes: Strategies to Decipher the 17 Gbp Hexaploid Bread Wheat Genome

Allohexaploid bread wheat (*Triticum aestivum*L., $2n=6\times=42$, AABBDD) is one of the world's major crops, which provides raw material for myriad industrial products and contributes essentially to livestock feeding and human nutrition accounting for approximately 20 % of daily consumed calories [1]. In particular with regard to global environmental, demographic, and economic changes, understanding genome structure, gene composition, and transcriptional regulatory mechanisms is crucial for the identification and improvement of agricultural and industrial important traits to ensure global food security [2]. However, the comprehensive analysis of the bread wheat genome and transcriptome has been substantially hampered by the absence of a suitable reference genome sequence. While ongoing improvement of next-generation sequencing (NGS) [3] technology allows the cost-efficient and rapid generation of sequence resources, large genome size (17.1 Gbp) [4] and high degree of repetitive elements (>80 %) [5] are a bottleneck for bioinformatic analysis and genome assembly. Moreover, allohexaploid bread wheat was formed by two independent hybridization events that brought together three diploid genomes (A, B, and D genome). As a consequence, homeologous sequences, especially in protein-coding regions, appear to be highly similar among each other [6]. The high similarity and redundancy in the pool of NGS whole-genome shotgun sequencing data impedes distinguishing homeologous sequence copies and leads to collapse of homeologous sequences when using traditional assembly concepts.

Two main approaches were used for the large-scale genome analysis of bread wheat. The first one combined low-coverage, long-read (454) whole-genome shotgun sequencing technology with a novel assembly concept on the basis of an orthologous gene framework [7]. Complementary, a second effort was undertaken by the International Wheat Sequencing Consortium (IWGSC) using flow-sorting technology to isolate DNA of individual chromosome-arms, which were subsequently sequenced using Illumina technology and assembled separately de novo [8].

2.1.1 Analyzing the Bread Wheat Genome Using Whole-Genome Shotgun Sequencing

To analyze the gene repertoire of the complex and highly repetitive bread wheat genome we developed and applied an orthologous group assembly (OA) strategy. This strategy uses low and medium coverage, long-read (Roche 454) whole genome shotgun data and can be readily applied to other, previously uncharacterized polyploid genomes. Homeologous gene copies from the three wheat

subgenomes tend to collapse in a standard de-novo assembly approach due to their overall high sequence similarity. A stringent assembly protocol was developed, which makes use of rare sequence polymorphisms, to generate sequences of distinct homeologous genes. Using this approach distinct gene copies encoded by the A, B, and D genomes could be distinguished and a total of 94,000–96,000 wheat genes were estimated [7].

The orthologous group assembly makes use of conserved sequence homology to closely related and—in part—fully sequenced plant organisms with smaller genome size and lower repeat content. Orthologous genes from multiple reference species were clustered and one representative gene model per group was selected as an orthologous group representative (OGR). Raw sequence reads from the NGS sequencing were aligned against each OGR defining “in-silico sequence capture bins” which were then independently assembled using stringent parameters. These assembly parameters were derived and evaluated from in-silico simulations of whole genome sequencing experiments. The resulting sub-assemblies were then used to estimate gene copy numbers in wheat and facilitate downstream analysis.

Defining an Orthologous Gene Set

Orthologous gene groups were computed with the OrthoMCL [9] software for the reference genomes of three grass genomes (*Brachypodium distachyon* [10], *Sorghum bicolor* [11], and *Oryza sativa* [12]) originating from different grass subfamilies plus publicly available barley full-length cDNAs [13]. One representative gene model was selected from each of the orthologous gene clusters.

Aligning Sequence Reads to Orthologous Gene Representatives

To effectively associate raw 454 sequence reads to the orthologous gene representatives, repetitive sequences had to be filtered out first. In the masking step, about 77 % of the raw sequence reads were removed, reducing complexity and search space significantly. Raw reads were aligned to orthologous gene representatives with BLASTX [14] using different alignment identity thresholds to reflect evolutionary distances between bread wheat and the reference species.

Generating Gene-Centric “Sub-assemblies” Using the Newbler Assembler

Individual assemblies were computed for all orthologous groups with the Newbler de novo assembly software [15]. The Newbler algorithm generates longer contigs based on overlaps between reads for which the minimum alignment identity (*mi*) needs to be defined in advance [16]. The minimum alignment identity parameter has a strong influence on the number of homeologous gene copies determined as very stringent *mi* values would overestimate copies due to sequencing errors whereas low *mi* values could cause homeologous gene copies to collapse in the assembly. As a consequence, two distinct methods were used to select for optimal *mi*

parameters in the assembly of raw reads mapped to orthologous groups representatives.

Estimating Appropriate Assembly Parameters

To determine optimal assembly parameters for the Newbler algorithm, we conducted two distinct simulations: (a) simulation of a whole-genome sequencing experiment for a diploid reference genome of similar genome size and repeat and gene content and (b) in-silico generation of a hexaploid gene set with similar sequence identity differences as observed for the homeologous genes in bread wheat.

Simulation of a Whole Genome Sequencing Experiment

The repeat-masked genome sequence of maize (ZMb73, version 5b.60 [17]) was used as a reference to simulate an orthologous group assembly with a finished, large genome and adjust assembly parameters based on observed and expected gene family sizes. We first determined gene family sizes from clustered maize protein sequences with the ones from the reference species used for the wheat analysis. From the resulting gene groups, all clusters with at least one maize gene and exactly one OGR were selected. The ratio between maize gene copies and OGR provides a reference value for the subsequent analysis.

After that, an in-silico dataset of raw 454 sequencing reads with fivefold genome coverage was generated from the maize reference assembly [18]. This dataset was then used to perform an orthologous group assembly according to the protocol described before testing minimum overlap identities between 97 % *mi* and 100 % *mi*. For each orthologous group and all *mi* parameters, the corresponding gene copy numbers for “simulated” maize was computed and compared to the observed “real-world” gene cluster compositions as determined in the OrthoMCL [9] clustering before. In consequence, this comparison allows estimating a best fit for the *mi* parameter, which approximates best the maize gene copy numbers for each cluster.

Simulation of a Polyploidy Gene Catalogue

Another evaluation of assembly parameters, which is complementary to the method outlined just before, also considers the effect of polyploidy on gene family size in a simulation of a whole genome sequencing experiment. Here, rice transcript sequences were aligned against the orthologous gene representatives, determining the rice copy number for each OGR. Each rice transcript which aligned to an OGR was then triplicated and random single nucleotide variants (SNVs) were introduced to simulate the sequence similarity of the homeologous genes in the polyploid wheat genome. Again, raw 454 sequencing reads with fivefold genome coverage were generated in-silico from the triplicated rice transcripts and mapped against their OGR. The mapped reads were assembled with varying minimum overlap identities between 97 % *mi* and 100 % *mi* and gene copy numbers were determined for each orthologous group.

Observed gene copy numbers widely differ depending on the minimum identity overlap parameters applied in the assembly of raw reads on OGRs. With both evaluation approaches, raw 454 reads coming from distinct gene copies were collapsed using 97 % *mi*, whereas 100 % *mi* (requiring perfectly conserved alignment overlaps) clearly results in an overestimation of the gene family sizes. Using a 99 % alignment identity threshold we observed an almost perfect agreement in the 1:1 relationship between the expected and observed gene family size in the maize simulation, as well as in the 1:3 relationship in the polyploidy simulation. Using this parameter in the assembly of reads mapped to OGRs allows for a compensation of sequencing errors, while maintaining distinct gene copies with high sequence similarity in coding regions.

Sub-genome Classification
of Bread Wheat Transcripts
Using a Machine Learning
Approach

Not only the large genome size complicates the analysis of the bread wheat genome but also its complex, allo-hexaploid genome structure. It is important to not only separate homeologous gene copies in bread wheat but also to classify coding sequences according to their subgenome origin (in wheat: *A*, *B*, and *D* subgenomes respectively). Applications of this classification include the design of subgenome specific markers and other breeding strategies as well as open questions in the evolution and domestication of bread wheat.

The orthologous group assembly approach (introduced above) uses a fivefold 454 whole genome sequence survey of the bread wheat genome to construct gene sub-assemblies, without being able to classify them into *A*, *B*, and *D* subgenome directly. In principle, a direct separation can only be facilitated with experimental/physical separation of individual chromosomes through flow cytometry [19] (*see* also next section). However, at the time of undertaking the analysis sorted sequences were only available for wheat linkage group 1 [5].

Therefore, an alternative approach was developed to assign wheat sub-assembly sequences to their subgenome affiliation. To facilitate classification, genome sequences were generated for the diploid progenitor of the wheat *D* subgenome, *Aegilops tauschii* [20], as well as for the close relative of the *A* subgenome *Triticum monococcum* (NCBI archive SRP004490.3), and cDNA sequence assemblies were produced for *Aegilops speltoides* (Trick & Bancroft, unpublished data), a member of the *Sitopsis* section to which the putative *B* genome donor has been proposed. Expecting that *A*-related sub-assemblies are more related to *T. monococcum* sequences, *D*-related sub-assemblies to *Ae. tauschii*, and *B*-related sub-assemblies to *Ae. speltoides*, sequence similarities of the sub-assemblies to each of these datasets would define and discriminate their homeologous origin.

Sequence similarities of each sub-assembly sequence to the wheat progenitor sequences were computed using BLAST [14]

and only sub-assemblies with hits to all three wheat progenitor sequences were considered for further classification. While the classification into A, B, or D subgenome derived transcripts seemed possible for many sub-assemblies using fixed similarity cutoffs, this approach showed unsatisfactory specificity for the majority of genes.

As a consequence, a number of different machine learning approaches were investigated and evaluated for their performance. A key factor is the identification and preparation of both a training and test data set. For that, chromosome sequences from the wheat linkage group 1, separated by subgenome using flow-sorting of chromosomes, were utilized [5]. Depending on highest sequence homology wheat sub-assemblies were assigned to the A, B, or D subgenome of wheat chromosome group 1. Finally, corresponding similarities to *T. monococcum*, *Ae. speltooides*, and *Ae. tauschii* sequences were identified for each sub-assembly to create a training set and to learn parameters for the genome-wide classification of the remaining wheat sequences.

A number of machine learning algorithms were tested with this training set such as Logistic Regression, Naive Bayes, Decision Trees, and Support Vector Machine algorithms from the WEKA package [21] and results were evaluated by stratified k-fold cross-validation. The best compromise between precision and recall was observed using Support Vector Machines (SVM). This trained SVM classifier was used to classify the set of wheat sub-assemblies into A-, B-, or D-related sequences. As a result, the homeologous subgenome relationships could be resolved for two-thirds of the sub-assemblies and have been assigned to either A-, B-, or D-subgenome with high classification probability. The remaining one-third was considered as unreliable predictions including, for example, cases with identical matches for a sub-assembly sequence to all three progenitor sequence sets [7].

2.1.2 Chromosome-Based Analysis of the Bread Wheat Genome

Generation of Chromosome(-arm) Sequence Resources

To overcome the challenges of genome-wide analysis in the bread wheat genome, one fundamental concept has been the reduction of overall genome complexity onto the level of individual chromosome arms [22]. Utilizing flow-cytometric sorting, wheat chromosome arms can be separated and purified with high accuracy [23, 24] providing a basis for chromosome-based construction of BAC libraries [25, 26] or high-throughput NGS sequencing [8, 27–32].

As only chromosome 3B can be sorted by size alone, the remaining chromosome arms were derived from double ditelosomic stocks of aneuploid bread wheat lines of hexaploid bread wheat cultivar “Chinese Spring” [23, 24, 33]. Within the IWGSC framework [8], the chromosome-based sequencing libraries were generated, sequenced with high-depth Illumina technology applying different mate-pair library sizes and de novo assembled. Although a large proportion of the repetitive sequences were collapsed and

the chromosomes only fragmentary assembled, the chromosomal survey sequences (CSS) represented more than 60 % of the sequence of the bread wheat genome. However, this data set constitutes an important step towards a high-quality reference genome sequence and a comprehensive understanding of the bread wheat genome. Moreover, it dramatically increases the resolution of current genomic analysis and provides valuable genomic sequence resource allowing the homeologous-specific annotation of (protein-coding) genes, structural comparisons and gene expression analysis in an important hexaploid cereal.

Consensus Gene Modeling and Gene Annotation

Structural annotation of functional, (protein-)coding genes is one of the basic steps of each genome analysis project. For the annotation of the CSS assembly, a homology-based (extrinsic) consensus gene modeling and annotation pipeline has been implemented utilizing high-quality protein annotations of related grass genomes barley [34], *Brachypodium* [10], rice [12] and sorghum [11], publically available full-length (fl) cDNA sequences [35] and a multi-organ RNA-sequencing library (Fig. 1).

Size and repetitivity of the bread wheat genome constitutes a major bottleneck in computation of spliced-alignments of query sequences against the assembled genome sequence. As part of PlantsDB the MIPS Repeat Element Database (mips-REDat [36]) encompasses a large amount of sequences of known repeat

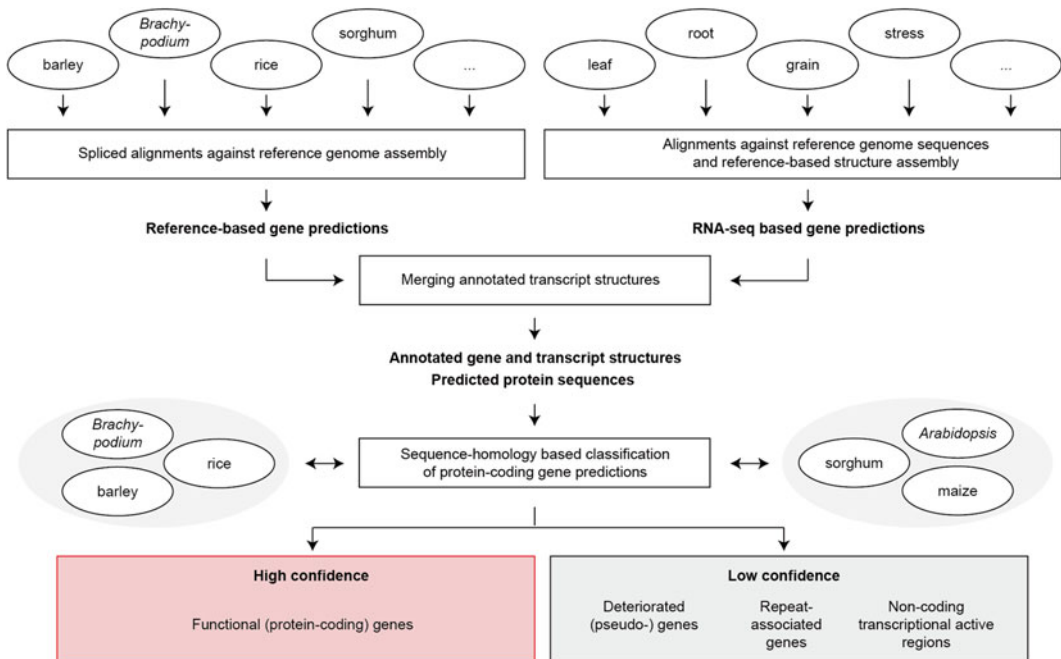


Fig. 1 Schematic workflow of the gene annotation for bread wheat

elements, which allows using nucleotide homology search to mask and filter for repetitive genomic regions. Besides considerable reduction of the search space and computational complexity, this pre-processing also avoids the false positive annotation of repeat-associated elements due to spurious seeds during the alignment of reference proteins.

In a three-step annotation protocol, first, wheat cDNA sequences and reference plant proteomes were aligned against the genome assembly. The resulting spliced-alignments were stringently filtered for transcript structures leading to truncated protein translations, interrupted by internal (nonsense) stop codons. These may represent non-function gene fragments or pseudogenes resulting from transposable element activity, a frequent feature of Triticeae genomes [5] or reflecting gene degeneration as a consequence of the polyploid evolution of bread wheat [37]. Transcript structures were separately assembled for the wheat transcriptome sequencing data resampling five organs. The previously defined annotation, which has been constructed using proteomes of the reference plant genomes, was supplied as backbone annotation to guide the assembly of the mapped short RNA-seq reads. This proved especially valuable for the detection and annotation of non- or only low expressed genes. Finally, a consensus gene and transcript set was created and, thereby, redundant transcript structures annotated in multiple tissues clustered based on shared intron boundaries.

However, genomic characteristics of the bread wheat genome, like the high abundance of gene fragments and pseudogenes [5, 7], as well as technical limitations, like split genes on multiple contigs in the CSS assembly, require further stringent post-processing of the obtained transcript structures. On the basis of the degree of conservation of peptide sequence homology between predicted wheat peptides and reference plant proteins, the annotated transcripts classify into low confidence (LC) and high confidence (HC) gene categories. While predicted genes of the HC class are most probable intact, functional protein-coding transcripts, the LC genes accumulate deteriorated gene (-fragments) or transcriptional active (noncoding) regions.

Novel Perspectives
for Homeologous-Specific
Genome and Transcriptome
Analysis

The established genome sequence assembly and gene annotation constitute a big step for the analysis of the bread wheat genome. Especially the separation of homeologous sequence copy enables to investigate the retention, loss or gain of genes, to screen genome-specific contributions to gene families of interest and to unravel phylogenetic relationships. Moreover, the CSS assembly is a suitable reference genome sequence for the application of high-throughput next generation sequencing technologies and to profile homeologous-specific transcriptional activity.

Protein Family Analysis

An important application for many gene-centric analyses is the genome-wide identification of protein families and orthologous groups shared with closely related species. In most cases, sequence homology-based Markov cluster (MCL) algorithms were utilized to group homeologous genes, orthologs, or very close paralogs [9, 38]. For example, these approaches allowed unraveling incongruous composition of protein families on a chromosome arm level in bread wheat suggesting inherited differences in gene families in the diploid progenitor genomes or evolutionary mechanisms acting differently on particular homeologous chromosomes, chromosome arms or chromosomal segments. With availability of more and more Triticeae genome sequences, which includes the diploid progenitor genomes *T. urartu* [39] and *Ae. tauschii* [40], comparative protein family clustering also unraveled the degree of gene loss and retention in the hexaploid genome and to identify genome- and lineage specific genes. Besides these genome-wide considerations, the bioinformatics analysis provided valuable resources for a targeted analysis of agricultural and industrial important wheat traits.

Homeologous Gene Triplets

Protein families often contain unbalanced numbers of genes from the A, B, and D genomes. The identification of homeologous genes, that are present as a single copy in each wheat subgenome, enables direct comparative analysis, like the evolutionary relationships, protein sequence divergence or differences in homeologous gene expression regulation. A conservative approach for the computation of “triplets” used pairwise bidirectional protein blast searches between the A, B, and D gene sets and led to the detection of more than 7000 triplets including >20,000 wheat HC genes, which were functionally annotated (gene ontology, PFAM domains, human readable gene descriptions).

Comparative Analysis of Diploid, Tetraploid, and Hexaploid Wheat Genomes

The discovery of single nucleotide sequence variations (SNVs) between bread wheat and related diploid and tetraploid genomes is particularly important for efficient marker development and breeding strategies. Favorable economics of NGS allowed to sequence representative genomes for the A (e.g., *T. urartu*, $2n=2\times=14$), B (e.g., *Ae. speltoidis*, $2n=2\times=14$) and D (*Ae. tauschii*, $2n=2\times=14$) genome lineages and tetraploid pasta wheat (*T. turgidum*, $2n=4\times=28$, AABB) and to obtain a set of draft genome assemblies. These draft genome assemblies prove to be particularly useful for a direct comparison against wheat coding sequences and to delineate SNVs. The superimposition of SNVs identified in pairwise comparisons between bread wheat and each subject genome allowed to delineate genome-specific or shared SNVs, which were used for reconstruction of phylogenetic relationships among wheat genomes of the A, B, and D genome lineages and to measure the impact of nucleotide substitutions on protein function by analysis

of synonymous and nonsynonymous nucleotide substitutions with Grantham score matrices.

Analysis of Homeologous Gene Expression

Analysis of wheat gene expression with both, microarray technology and high-throughput mRNA-sequencing (RNA-seq), until recently was restricted to a global perspective since discrimination of homeologous transcripts and genome-assignment was limited. With the availability of the CSS wheat assembly, RNA-seq reads can now be aligned against the genomic sequences and, consequently, assigned to their genome-of-origin. However, stringent filtering of ambiguously mapped reads (i.e., more than one mapping location) and filtering of paired-end reads with dissonant alignments (i.e., to sequences of different chromosomes) is necessary for an accurate computation of gene expression. Besides the global characterization of the wheat transcriptome, in particular comparative expression analysis of gene expression between homeologous genes forming triplets permits to elucidate genome-specific differences and to identify significantly differential expressed homeologs. This might help to gain a detailed understanding of gene associations and contributions of individual wheat genomes to protein abundances in different organs, tissues and cell types or under specific environmental influences.

2.2 The GenomeZipper Concept Facilitates the Anchoring and Ordering of Genes in Complex Grass Genomes

Whole genome shotgun sequencing in large and complex genomes still faces the challenge of downstream assembly and positional ordering of sequence contigs along chromosomes. While this challenge has largely been solved for smaller and less complex genomes in e.g., Triticeae highly fragmented assemblies with 100,000's of individual assemblies persist. On the other hand pronounced synteny among the grasses, embracing evolutionary distances of 50 million years, has been demonstrated [41]. In the GenomeZipper concept syntenic conservation among grasses over larger evolutionary distances, available high-density marker maps and the availability of fully sequenced reference genomes are exploited to deduce a virtual, approximated gene order (Fig. 2). Albeit limitations are apparent the virtual ordering can be seen as an intermediate step towards the goal of completely sequenced (and thus ordered) genomes and chromosomes.

The approach uses a molecular marker scaffold along with shotgun sequence data and reference genomes. For barley, rye and wheat the fully sequenced genomes of *Oryza sativa* (rice) [12], *Brachypodium distachyon* (Brachypodium) [10] and *Sorghum bicolor* (sorghum) [11] were routinely used. Since chromosome sorting from aneuploidy deletion or substitution lines is routinely used for chromosomal shotgun/survey sequencing (CSS) in a first step the corresponding syntenic regions in the model/reference genomes are deduced. Shotgun reads or assemblies from the respective chromosomes under investigation are compared against

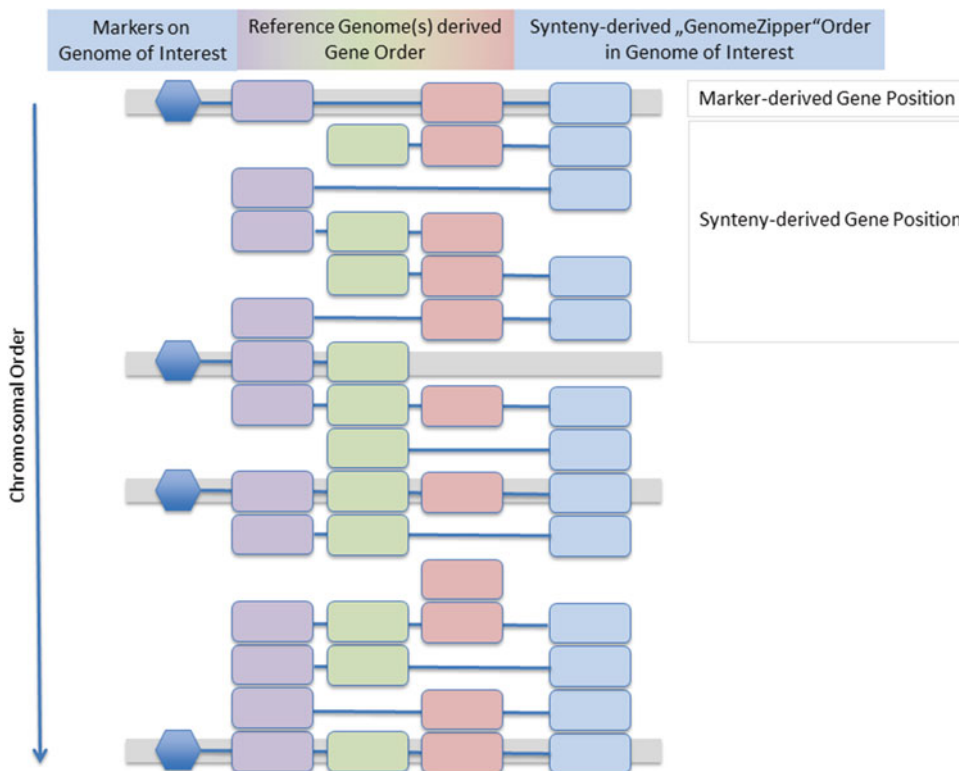


Fig. 2 Schematic overview about the GenomeZipper concept. The GenomeZipper approach exploits pronounced synteny among grass genomes and combines a marker framework with a positional information of orthologous genes in fully sequenced reference genomes. Marker data are connected to corresponding (orthologous) genes in the reference genomes using best bidirectional hit criteria (BBH). Scaffold intervals between marker tagged anchor points are derived and are used as a template to order and associate BBH hits from shotgun data/assemblies in the species of interest. The resulting order is synteny informed and can be seen as a virtual or approximative gene order

the repeat masked model genomes (BLAST). Corresponding regions attract most of the similarity matches and are identified using visual inspection and match density measuring metrics. These syntenic regions are selected and interlinked with a given marker scaffold of the genome under investigation. Markers are associated with corresponding (likely orthologous) genes in the reference genome(s) using a best reciprocal match (best bidirectional hit (BBH)) criterion and thus give a direct and sequence based interconnection between the genome under investigation and the respective reference genome(s). This does not only define syntenic intervals by the anchor points and the genes located in these in the intervals on the respective model genome(s) but also interlaces the corresponding and overlapping regions among the different reference genomes used. Anchoring and, derived from the sequential order of genes in fully sequenced reference genomes, virtual

ordering of genes in the inter-marker space subsequently allows to scan the remaining CSS for BBH genes and position them along the reference genome directed template.

Meanwhile the GenomeZipper concept has been applied for a variety of cereal genomes, namely barley, wheat, rye, *Aegilops* and *Lolium* [20, 30, 42–45]. The resulting data generate large tables of virtual order, coordinates and sequences along the chromosomes that due to complexity and size exceed the size limitations of traditional publications. Consequently browsable and navigatable views of the various Genome Zipper derived orders for genomes and chromosomes are also made available through the PlantsDB information portal at <http://mips.helmholtz-muenchen.de/plant/triticeae/genomes/index.jsp>. The views enable various search axis using keyword searches, searches along the physical/genetic axis as well as homology based searches and candidate gene searches.

3 Methods

3.1 PGSB/MIPS PlantsDB: A Resource for the Management and Comparative Analysis of Plant Genome Data

The PGSB/MIPS PlantsDB system has been designed as an information resource for plant genome data. Its aim is to structure and communicate plant genomic data and assist the comparative analysis of both model and crop plant genomes. PlantsDB currently hosts dedicated instances for the plant model organisms *Arabidopsis thaliana*, *Oryza sativa* (Rice), *Medicago truncatula*, *Zea mays* (Maize), *Solanum lycopersicum* (Tomato) and many more. To accommodate data and analysis results from the complex genomes of barley, wheat and rye, an instance specifically attributed to Triticeae was generated within PlantsDB.

Many plant genome datasets stored within PGSB/MIPS PlantsDB were generated in collaborative efforts. Ongoing and long-term involvement in numerous international plant genome projects results in highly curated datasets such as gene predictions, repetitive element libraries or in the finishing and anchoring of complex plant genome sequences (e.g., wheat and barley). These curated datasets represent a fundamental data resource for experimental plant biologists and breeders.

Beside information on individual genetic elements, their physical and functional properties and information about the underlying nucleotide or amino acid sequence, more and more combinatorial and comparative queries become important to address more complex scientific questions. To assist these, specialized tools and interfaces were developed or integrated in PlantsDB, including the synteny browser CrowsNest and an expression browser.

3.1.1 PlantsDB System Architecture and Design

Datasets generated within modern plant genome analyses are typically complex and datatypes may be heterogenous and hard to integrate with others. The PlantsDB database architecture has

been set up with a modular data structure to accommodate newly emerging datatypes and to be able to establish new connections between existing and/or new datatypes. Incorporation of controlled vocabulary, ontologies (such as Gene Ontology, GO) [46] and the definition of standardized data (exchange) formats accounts for seamless data integration although data integration and management often still requires significant manual curation efforts in practice.

The core data is stored in an ORACLE relational database management system. A modular approach has been chosen for the implementation of MIPS plant genome resources. To manage these data modules in a component-oriented manner, a multi-tier architecture following the J2EE standard has been implemented.

The core of the system is constituted by a flexible and generic data model for the representation of genome sequence and annotation. Three basic entities are defined: *Clone*, *Contig* and *GeneticElement*. *Clones* store sequence and attached information that relates to a physical clone. To assemble a representation of a genome sequence, clone sequences are processed to remove overlaps and redundancy, ambiguous sequence or vector contamination, and then stored as *Contigs*. The *Contig* data module also stores information on how to assemble individual clones into longer contigs and pseudomolecules representing whole chromosomes.

The third data module, *GeneticElement*, stores all genetic elements anchored on the genome sequence: Protein coding genes, noncoding RNAs, repeats, sequenced markers, transposons, etc. GeneticElements can consist out of subelements, e.g., exons, introns, UTRs or domains that constitute a particular *GeneticElement*. This way, all *GeneticElements* associated to a gene (promoter, transcript, alternative transcripts, regulatory elements, cDNA matches, etc.) can be identified through a single group entry.

For every plant species, a separate physical instance of all three generic data modules is created to ensure scalability and separation of name spaces.

Plant genome databases and resources are non-isolated fields but data, species, tools/interfaces, or objectives potentially complement or intersect each other. To facilitate database cross talk and help users in aggregating distributed and/or heterogenous data, several strategies were proposed over the last few years. PlantsDB implements webservices for the remote access to its data and services as developed by the BioMOBY consortium (www.biomoby.com) [47, 48]. Moreover, PlantsDB is part of the transPLANT consortium, an EU initiative to facilitate trans-national infrastructure and interconnection of plant genome data (www.transplantdb.eu). Within that project, an international plant genome resource registry is maintained by PlantsDB and cross-search functionality was implemented between major European plant genome databases.

3.1.2 *PlantsDB Analysis Tools, Web Interface, and Data Retrieval*

All datasets stored within PlantsDB are made accessible for search, download, and (comparative) genome analyses.

To browse data, the user can navigate in a genome-oriented way. Assuming one would e.g., start from the chromosome list, all contigs anchored to each chromosome can be retrieved. A contig report contains detailed information on the entry as well as links to sequence, external database records, a list of annotated genetic elements or a graphical viewer. The genetic element list links to reports on the protein genes or other features. Sequences can be viewed and downloaded as HTML, XML or FASTA format. For protein coding genes, unspliced, spliced (transcript) and coding DNA sequences as well as protein sequences are available. Moreover, cross-references in the reports allow easy access to entries in external databases associated with the entry (e.g., relevant literature in PubMed).

Alternatively, complete lists of all sequenced contigs, all genetic elements or all elements of a selected type are available for browsing. To visualize and browse genetic elements on a specified contig or chromosome, Gbrowse [49] views have been integrated for some species.

Search options include search by name, free text or sequence. The free text search option allows inspection of the content of all text fields, and it is available for individual genomes or across all databases. BLAST [14] is used as a homology search engine. The target databases for similarity searches include genomic sequences (such as scaffolds or chromosomes) and genetic elements (such as coding sequences).

The download section of PlantsDB provides access to various data downloads via the FTP protocol. This includes FASTA-formatted--> sequence files for genomic sequence and protein-coding genes.

To address structural genome characteristics of plants such as conserved gene order between many monocot plant genomes, custom tools such as the *CrowsNest* synteny viewer were developed and integrated with other views such as gene reports or precalculated gene family results.

MIPS PlantsDB can be accessed at <http://mips.helmholtz-muenchen.de/plant/genomes.jsp>.

3.1.3 *CrowsNest: A Tool to Explore and Visualize Syntenic Relationships in Plants*

CrowsNest is a synteny viewer that allows comparisons on the basis of genetically and physically anchored genomes. It enables to visualize syntenic segments, orthologous and homologous gene pairs for selected plant genome comparisons (Table 1). The CrowsNest tool is online accessible at: <http://mips.helmholtz-muenchen.de/plant/crowsNest/>.

From Genome-Wide Comparison to Gene-Gene Comparisons

CrowsNest offers different levels of detail when comparing up to three genomic datasets (one target and one or two reference datasets). To illustrate these levels an example is used (Fig. 3) in which

Table 1**Genome comparisons online available in CrowsNest (May 2014)**

Genome A	Genome B
<i>Brachypodium distachyon</i>	<i>Oryza sativa</i>
<i>Brachypodium distachyon</i>	<i>Sorghum bicolor</i>
<i>Oryza sativa</i>	<i>Sorghum bicolor</i>
<i>Hordeum vulgare</i>	<i>Brachypodium distachyon</i>
<i>Hordeum vulgare</i>	<i>Aegilops tauschii</i>
<i>Hordeum vulgare</i>	<i>Oryza sativa</i>
<i>Phoenix dactylifera</i>	<i>Elaeis guineensis</i>
<i>Capsicum annuum</i>	<i>Solanum lycopersicum</i>

the *Brachypodium distachyon* (target) genome is compared with the *Oryza sativa* (reference) genome.

All Target Versus All Reference Chromosomes

The highest level of detail is displayed as a circular layout (Fig. 3a, b) showing a comparison of all target chromosomes with the respective syntenic regions on the reference chromosomes. This includes orthologs and homologs shared between the genomes.

In Fig. 3a the target chromosomes (here: Bd1–Bd5) are all consecutively displayed. Syntenic regions to the reference chromosomes are mapped onto them and are color-coded for better differentiation. For each target chromosome the distribution of features (e.g., introns, exons, transposable elements) is highlighted in the innermost circle if this data is available. In an optional view paralogs can be displayed, and the Z-score [44] as a measure for synteny at a certain region on the chromosome can be displayed.

In Fig. 3b reference (Os1–Os12) and target (Bd1–Bd5) chromosomes are all displayed in a circular layout and syntenic regions between them are displayed as ribbons. The ribbons are color-coded according to the reference chromosomes for better differentiation.

The syntenic regions between the target and reference genomes can also be displayed in a linear layout (Fig. 3c). The target chromosomes (Bd1–Bd5) are linearly aligned and syntenic regions are displayed as tracks aside of them. The tracks are again color-coded.

Single Target Against Multiple Reference Chromosomes

For a single target chromosome, a more detailed view can be selected in which this chromosome is compared against chromosomes from the reference genome(s). In this view different regions can be displayed: syntenic, orthologous and homologous (if

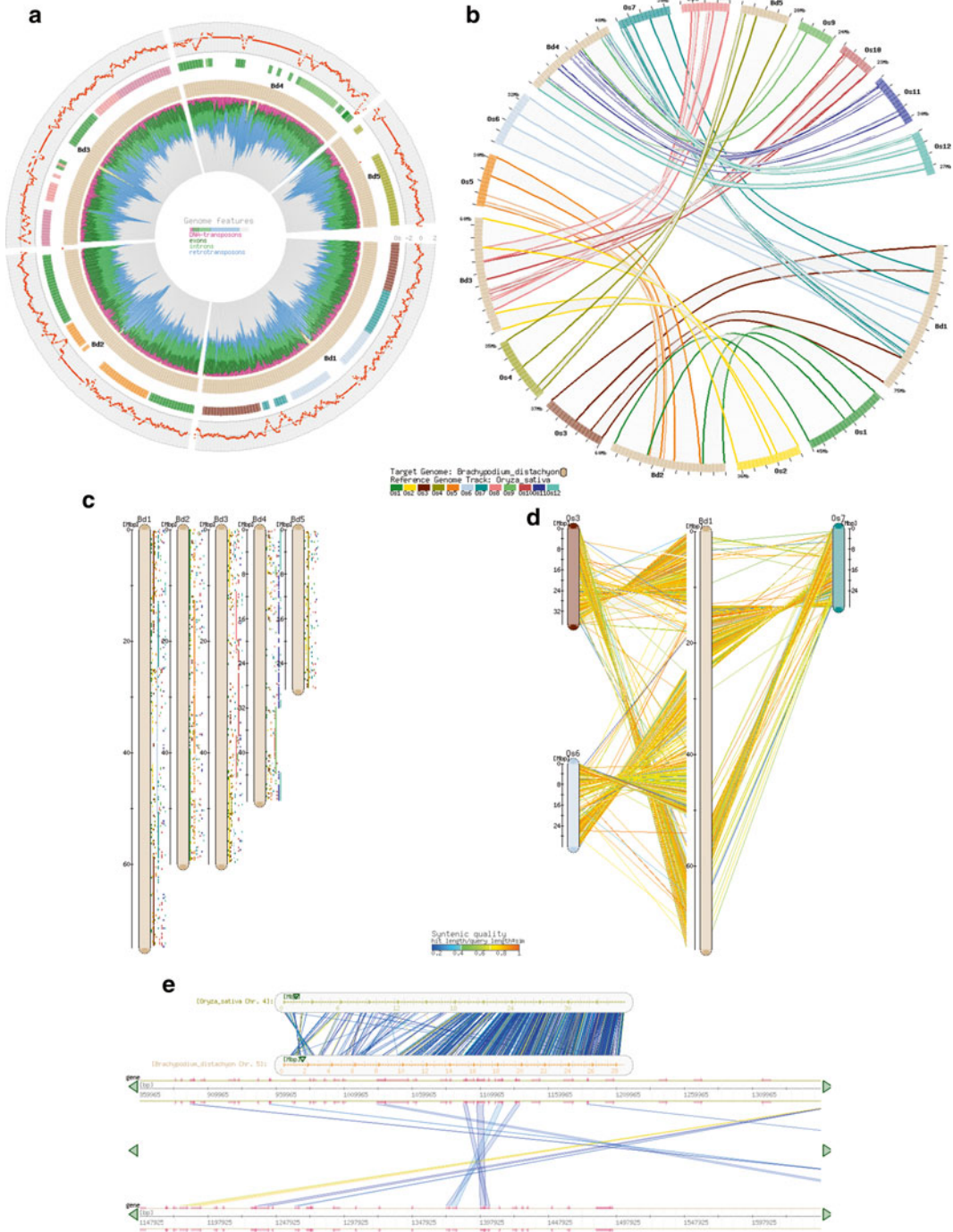


Fig. 3 Different levels of detail accessible via the CrowsNet tool, here comparing the *Brachypodium distachyon* (target) genome with the *Oryza sativa* (references) genome datasets (a). On the basis of orthologous glue pairs, (b) and (c) depicts the distribution of orthologous genes by comparing all chromosomes (here: bd1). (e) highlights micro-synteny in a selected region between chromosome 1 in *Brachypodium* and chromosome 7 in rice

available). Figure 3d shows an example in which the target chromosome (Bd1) is compared against multiple reference chromosomes (Os3, Os6, Os7). Syntenic regions are shown as lines between the chromosomes and the quality of the syntenic relationship is represented by the color of the lines. The color gradient ranges from blue (low quality) to orange (high quality).

Comparison on Gene Level

CrowsNest further allows to inspect the genomic context of a specific gene residing in the region of interest (Fig. 3e). If a target and a reference chromosome share a relevant amount of orthologous genes, a comparison on a portion of these two chromosomes can be performed. This is done by selecting a linkage between target and reference chromosomes in the previous view. The quality of the syntenic relationship is represented by colored links between them. Zooming and panning functions allow a further inspection of the selected region. From there, links to MIPS PlantsDB are provided to obtain additional information about the gene (e.g., sequence information, gene family assignment and anchoring information).

3.2 RNASeq-ExpressionBrowser

RNA-seq is a Next-Generation Sequencing [3] technology that provides insights into the RNA present at a defined time. It also allows drawing conclusions on e.g., alternative splicing and the abundance of alternative splice forms. Broadly applied RNA-seq pipelines for handling RNA data like Tophat [50] and Cufflinks [51] generate huge amounts of data, which imposes severe challenges (and limitations) on communication and sharing of the results. To facilitate this, we have developed the RNASeq ExpressionBrowser [52], a web-based tool for the search and visualization of RNA-seq expression data (accessible at <http://mips.helmholtz-muenchen.de/plant/RNASeqExpressionBrowser>).

The tool is implemented as a portable stand-alone software platform that enables incorporation and visualization of RNA-seq derived expression data for non-genome wide scans but for visualization and inspection of genes and regions of interest.

It allows searching for genes either by domain annotation, sequence similarity or gene list, and returns its results in form of detailed reports.

The RNASeqExpressionBrowser provides three different ways to access the processed RNA-seq datasets: it allows either the wildcard-search based on keywords, a BLAST search against the project associated sequences, or a search based on a gene list.

3.2.1 Annotation: Keyword Search

The search methods are displayed in Fig. 4. For the search via gene identifier, the gene name can be queried. Additionally, it is possible to search for genes using a prefix followed by the wildcard character "%". The keyword search field allows queries based on the gene

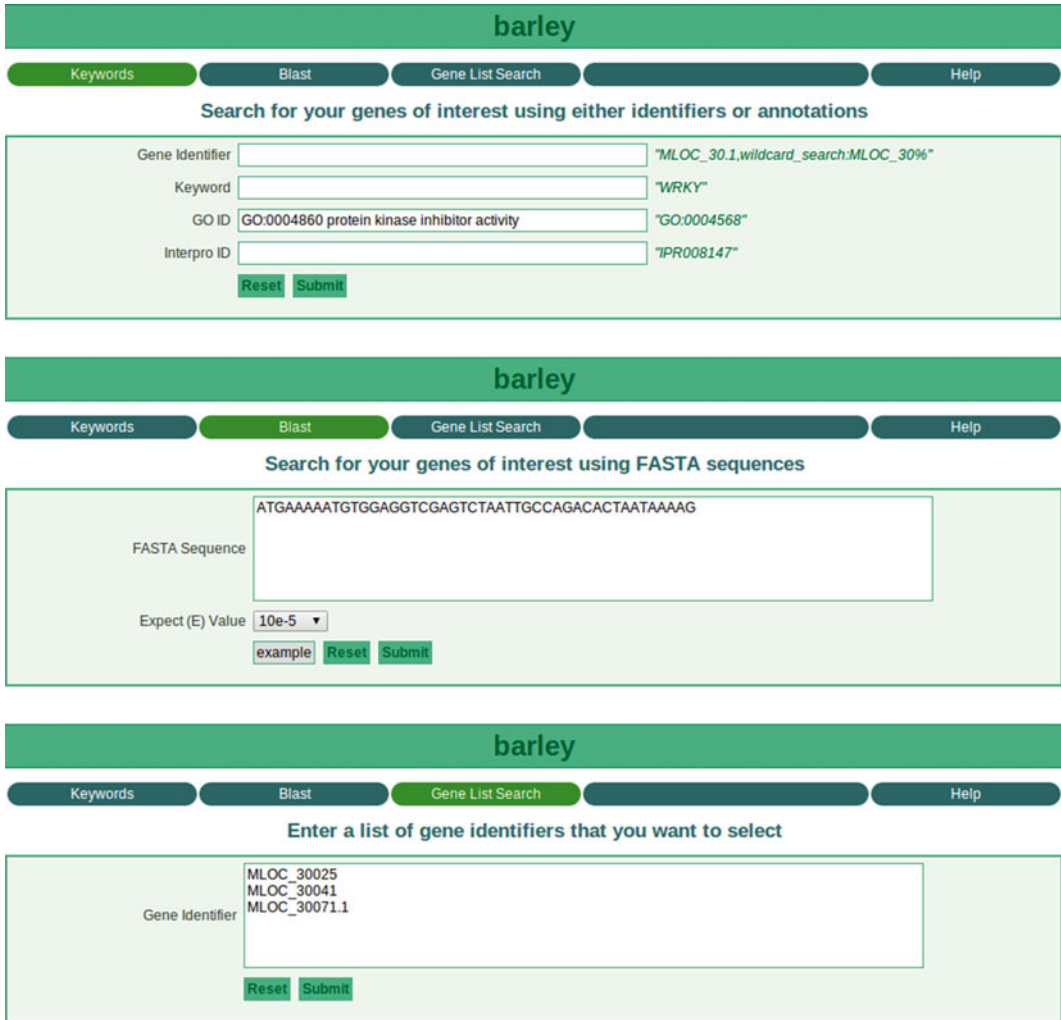


Fig. 4 Entry page of the RNASeqExpressionBrowser (a). Keyword search, allows to search by gene identifiers, keywords, or domain information. (b) A BLAST sequence search is provided where a threshold for the Expect (E) Value can be selected. (c) Also a search by gene identifiers is provided

description. Suggestions are provided via an autocomplete function, when at least three characters were provided by the user.

We also allow a Gene ontology (GO) term search. The Gene Ontology Consortium [46] aims at standardizing the representation of gene and gene product attributes across species and databases. The search is based on providing the ID (e.g., GO:0006556) rather than descriptive free-text (e.g., ‘S-adenosylmethionine biosynthetic process’), but RNASeqExpressionBrowser provides in addition an ID-to-term mapping. The search makes use of the GO hierarchy. Therefore, when searching a very general term, this can lead to increased search time. Additionally, other domain

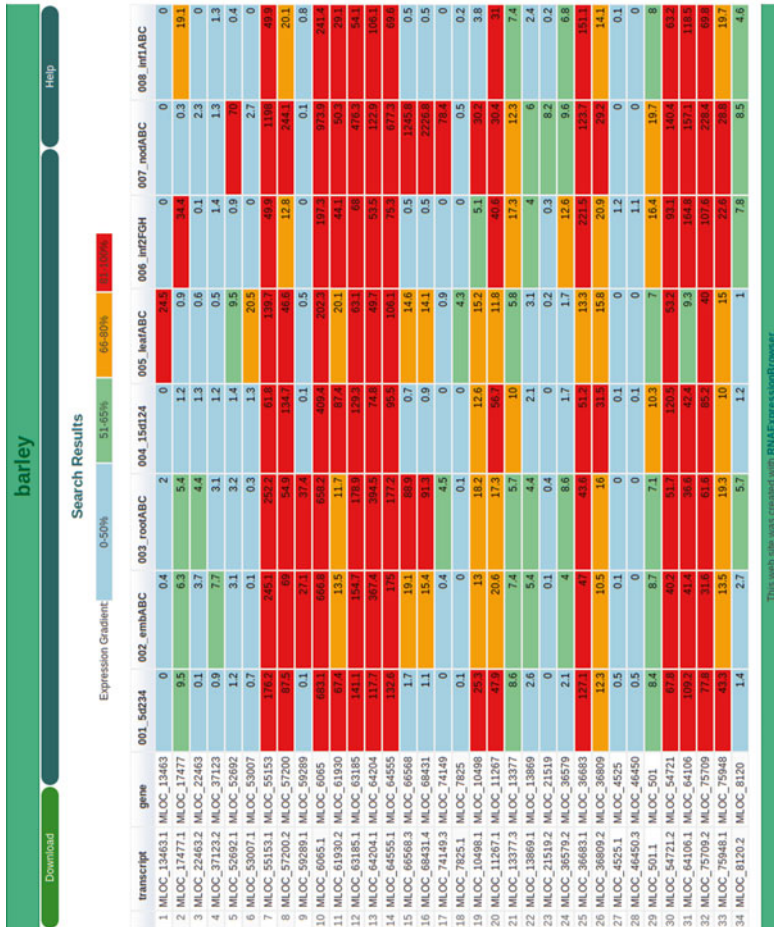
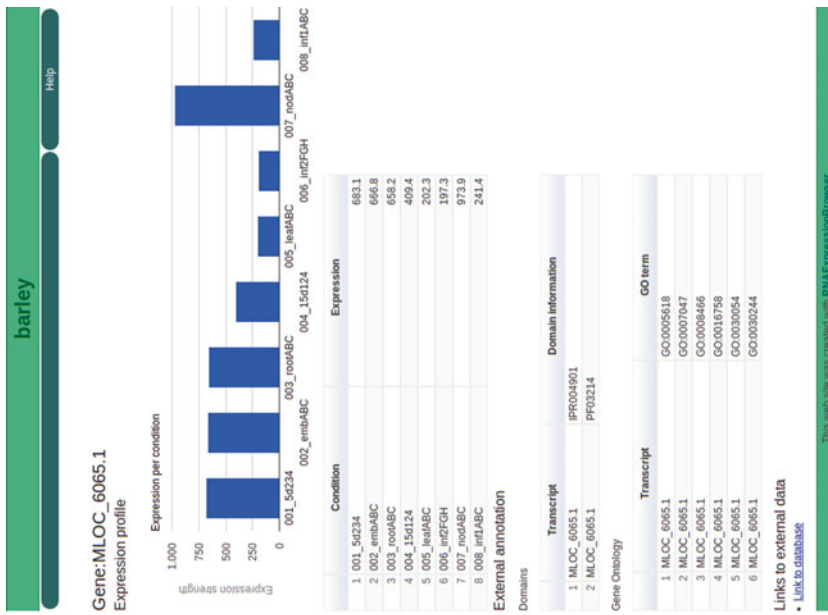


Fig. 5 (Left): Result page showing all transcripts annotated with a certain GO term (GO: 0051273). Each line represents a transcript. Red-colored blacks represent highly expressed transcripts, blue represent lowly expressed genes. After selecting a specific transcript, further gene details can be obtained by clicking on the gene ID. (Right): Detailed result page showing the expression profile for a selected transcript (here: MLOC_6065.1)

information can be integrated and searched. For other domain information the search term has to fully match to the provided domain terms (e.g., ‘IPR008147’).

3.2.2 Sequence Similarity

RNASeqExpressionBrowser provides a nucleotide BLAST search against all project associated sequences. This allows searching multiple nucleotide sequences against the project associated sequences. When matches were found, the search result reports the sequence identity, the match length and the underlying BLAST score. In addition, a link to the expression profile is included.

Search results: Report page

In the RNASeqExpressionBrowser, the results are presented as different views—a tabular overview (Fig. 5) listing all genes covered by the respective analysis. Upon selection of individual genes of interest a detailed view on the expression characteristics as well as structural and functional annotation information can be collated and summarized.

Genes in the tabular search results view can be reordered by selecting the corresponding column header. In the detailed view expression and annotation information for a selected gene are given. In the detailed view, the expression information is displayed both as a bar chart and in tabular form. Additional annotations, e.g., domains and gene ontology annotations are displayed. Links to external databases can be integrated.

References

1. Lobell DB, Schlenker W, Costa-Roberts J (2011) Climate trends and global crop production since 1980. *Science* 333(6042):616–620
2. Godfray HCJ et al (2010) Food security: the challenge of feeding 9 billion people. *Science* 327(5967):812–818
3. Matsumoto T et al (2005) The map-based sequence of the rice genome. *Nature* 436(7052):793–800
4. Eilam T et al (2007) Genome size and genome evolution in diploid Triticeae species. *Genome* 50(11):1029–1037
5. Wicker T et al (2011) Frequent gene movement and pseudogene evolution is common to the large and complex genomes of wheat, barley, and their relatives. *Plant Cell* 23(5):1706–1718
6. Mochida K, Yamazaki Y, Ogihara Y (2004) Discrimination of homoeologous gene expression in hexaploid wheat by SNP analysis of contigs grouped from a large number of expressed sequence tags. *Mol Genet Genomics* 270(5):371–377
7. Brenchley R et al (2012) Analysis of the bread wheat genome using whole-genome shotgun sequencing. *Nature* 491(7426):705–710
8. International Wheat Genome Sequencing Consortium (2014) A chromosome-based draft sequence of the hexaploid bread wheat (*Triticum aestivum*) genome. *Science* 345(6194):1251788
9. Li L, Stoeckert CJ, Roos DS (2003) OrthoMCL: identification of ortholog groups for eukaryotic genomes. *Genome Res* 13(9):2178–2189
10. International Brachypodium Initiative (2010) Genome sequencing and analysis of the model grass *Brachypodium distachyon*. *Nature* 463(7282):763–768

11. Paterson AH et al (2009) The Sorghum bicolor genome and the diversification of grasses. *Nature* 457(7229):551–556
12. International Rice Genome Sequencing Project (2005) The map-based sequence of the rice genome. *Nature* 436(7052):793–800
13. Matsumoto T et al (2011) Comprehensive sequence analysis of 24,783 barley full-length cDNAs derived from 12 clone libraries. *Plant Physiol* 156(1):20–28
14. Altschul SF et al (1990) Basic local alignment search tool. *J Mol Biol* 215(3):403–410
15. Margulies M et al (2005) Genome sequencing in microfabricated high-density picolitre reactors. *Nature* 437(7057):376–380
16. Miller JR, Koren S, Sutton G (2010) Assembly algorithms for next-generation sequencing data. *Genomics* 95(6):315–327
17. <http://www.maizegdb.org>
18. Richter DC et al (2008) MetaSim: a sequencing simulator for genomics and metagenomics. *PLoS One* 3(10), e3373
19. Dolezel J et al (2012) Chromosomes in the flow to simplify genome analysis. *Funct Integr Genomics* 12(3):397–416
20. Luo MC et al (2013) A 4-gigabase physical map unlocks the structure and evolution of the complex genome of *Aegilops tauschii*, the wheat D-genome progenitor. *Proc Natl Acad Sci U S A* 110(19):7940–7945
21. Frank E et al (2004) Data mining in bioinformatics using Weka. *Bioinformatics* 20(15):2479–2481
22. Gill BS et al (2004) A workshop report on wheat genome sequencing: International Genome Research on Wheat Consortium. *Genetics* 168(2):1087–1096
23. Vrana J et al (2000) Flow sorting of mitotic chromosomes in common wheat (*Triticum aestivum* L.). *Genetics* 156(4):2033–2041
24. Vrana J et al (2012) Flow cytometric chromosome sorting in plants: the next generation. *Methods* 57(3):331–337
25. Safar J et al (2010) Development of chromosome-specific BAC resources for genomics of bread wheat. *Cytogenet Genome Res* 129(1–3):211–223
26. Choulet F et al (2014) Structural and functional partitioning of bread wheat chromosome 3B. *Science* 345(6194):1249721
27. Berkman PJ et al (2011) Sequencing and assembly of low copy and genic regions of isolated *Triticum aestivum* chromosome arm 7DS. *Plant Biotechnol J* 9(7):768–775
28. Berkman PJ et al (2012) Sequencing wheat chromosome arm 7BS delimits the 7BS/4AL translocation and reveals homoeologous gene conservation. *Theor Appl Genet* 124(3):423–432
29. Ma J et al. (2013) Sequence-based analysis of translocations and inversions in bread wheat (*Triticum aestivum* L.). *Plos One* 8(11)
30. Hernandez P et al (2012) Next-generation sequencing and syntenic integration of flow-sorted arms of wheat chromosome 4A exposes the chromosome structure and gene content. *Plant J* 69(3):377–386
31. Belova T et al (2013) Integration of mate pair sequences to improve shotgun assemblies of flow-sorted chromosome arms of hexaploid wheat. *BMC Genomics* 14:222
32. Tanaka T et al (2014) Next-generation survey sequencing and the molecular organization of wheat chromosome 6B. *DNA Res* 21(2): 103–114
33. Sears E, S.L. (1978) The telocentric chromosomes of common wheat. In: Proceedings of 5th international wheat genet symposium, 1978, p 389–407
34. Mayer KFX et al (2012) A physical, genetic and functional sequence assembly of the barley genome. *Nature* 491(7426):711–6
35. Mochida K et al (2009) TriFLDB: a database of clustered full-length coding sequences from triticeae with applications to comparative grass genomics. *Plant Physiol* 150(3): 1135–1146
36. Nussbaumer T et al (2013) MIPS PlantsDB: a database framework for comparative plant genome research. *Nucleic Acids Res* 41(Database issue):D1144–D1151
37. Wendel JF (2000) Genome evolution in polyploids. *Plant Mol Biol* 42(1):225–249
38. Enright AJ, Van Dongen S, Ouzounis CA (2002) An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res* 30(7):1575–1584
39. Ling HQ et al (2013) Draft genome of the wheat A-genome progenitor *Triticum urartu*. *Nature* 496(7443):87–90
40. Jia J et al (2013) *Aegilops tauschii* draft genome sequence reveals a gene repertoire for wheat adaptation. *Nature* 496(7443):91–95
41. Bolot S et al (2009) The ‘inner circle’ of the cereal genomes. *Curr Opin Plant Biol* 12(2): 119–125
42. Martis MM et al (2013) Reticulate evolution of the rye genome. *Plant Cell* 25(10):3685–3698
43. Mayer KF et al (2011) Unlocking the barley genome by chromosomal and comparative genomics. *Plant Cell* 23(4):1249–1263

44. Mayer KF et al (2009) Gene content and virtual gene order of barley chromosome 1H. *Plant Physiol* 151(2):496–505
45. Pfeifer M et al (2013) The perennial ryegrass GenomeZipper: targeted use of genome resources for comparative grass genomics. *Plant Physiol* 161(2):571–582
46. Ashburner M et al (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet* 25(1):25–29
47. Wilkinson MD, Links M (2002) BioMOBY: an open source biological web services proposal. *Brief Bioinform* 3(4):331–341
48. Wilkinson M et al (2005) BioMOBY successfully integrates distributed heterogeneous bioinformatics Web Services. The PlaNet exemplar case. *Plant Physiol* 138(1):5–17
49. Stein LD et al (2002) The generic genome browser: a building block for a model organism system database. *Genome Res* 12(10):1599–1610
50. Trapnell C, Pachter L, Salzberg SL (2009) TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics* 25(9):1105–1111
51. Trapnell C et al (2010) Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat Biotechnol* 28(5):511–515
52. Nussbaumer T et al (2014) RNASeqExpressionBrowser—a web interface to browse and visualize high-throughput expression data. *Bioinformatics* 30(17):2519–2520

Chapter 9

MaizeGDB: The Maize Genetics and Genomics Database

Lisa Harper, Jack Gardiner, Carson Andorf, and Carolyn J. Lawrence

Abstract

MaizeGDB is the community database for biological information about the crop plant *Zea mays*. Genomic, genetic, sequence, gene product, functional characterization, literature reference, and person/organization contact information are among the datatypes stored at MaizeGDB. At the project's website (<http://www.maizegdb.org>) are custom interfaces enabling researchers to browse data and to seek out specific information matching explicit search criteria. In addition, pre-compiled reports are made available for particular types of data and bulletin boards are provided to facilitate communication and coordination among members of the community of maize geneticists.

Key words Maize, Database, Genetics, Genomics, Genome, Model organism database

1 Introduction

MaizeGDB is the repository for and interface to maize biological data. Many diverse types of public data can be found at MaizeGDB, such as: DNA sequence, genome assemblies, genetic maps, cytogenetic maps, haplotype maps, genetic mapping panels, genes and other loci, gene models, transcripts, ESTs, mutations, alleles, stocks, QTLs, SNPs, BACs, probes, gene products, proteins, pathway data, microarray data, expression atlas, RNAseq, microRNAs, insertion elements and stocks, images of mutants, images of gel patterns, references, people, organizations, tutorials, curated list of maize projects and resources, and community information such as history of the community. Formed in 2002 [1] by the fusion of two databases MaizeDB [2, 3] and ZmDB [4]; MaizeGDB is now a sequence centric one stop shop for maize biological data [5]. The MaizeGDB also provides a full featured Genome Browser which includes many custom tracks [5]. Data enters MaizeGDB in one of several ways: (1) Information from primary literature is entered via manual curation, (2) Data provided directly from maize researchers assisted by MaizeGDB, and (3) Data provided by other databases such as NCBI, Gramene, and PlantGDB.

MaizeGDB personnel work with generators of large datasets to get the data into a format that can be served and accessed at MaizeGDB. In addition to collecting, storing and making available maize data, workers at MaizeGDB also develop custom tools to interact efficiently with the truly enormous and complex datasets now available. “Big Data” is the catchphrase used to describe information sets that have extreme volume, variety, velocity, and complexity (reviewed at <http://blogs.sap.com/analytics/2012/04/11/big-data-for-small-companies/>). Traditional and even current data management applications are not designed to handle this data load. As it is used currently, “Big Data analysis” conveys the concept of extreme information management. Although large datasets are the aspect of Big Data management that is most obvious, large (and often fairly simplistic) datasets like genome-wide SNP datasets are actually the least interesting to serve and convey the least biologically relevant meaning. By comparison, the complexity aspect of Big Data management involves recapitulating biological meaning and is an aspect of the emerging discipline that MaizeGDB was a forerunner in addressing well. The depth and breadth of data at MaizeGDB continue to be a unique aspect of the data resource relative to other online repositories of biological information.

In addition to serving maize data and tools for interacting with the information, MaizeGDB personnel also provide services to the maize research community. Bulletin boards for news items, information of interest to cooperators, a curated list of maize projects and resources that focus on the scientific study of maize, an editorial board’s recommended reading list, and educational outreach items are among the webpages made available through the MaizeGDB site (*see* Table 1). In addition, workers at MaizeGDB provide technical support for the Maize Genetics Executive Committee and the Annual Maize Genetics Conference.

Information about the history of MaizeGDB and the technical aspects of project’s operation are described elsewhere [1, 5–8]. Reported here are the types of data and tools that are made available at MaizeGDB, some generalized search strategies that can be applied across various datatypes, and some specialized example usage cases. Mechanisms for adding data to the database also are described in detail.

2 Materials

This section lays out in detail the types of data stored at MaizeGDB.

2.1 Genomic Sequence Data

One item that relates directly to MaizeGDB’s mission is to serve as the long term steward of whole genome sequence assemblies from any *Zea mays* subspecies and inbreds. We house or maintain current links to:

Table 1
Bulletin boards and static pages

Page title and web address	Content description
News column http://www.maizegdb.org/	News bulletins are displayed in the right margin. Older items are accessible through a link near the bottom
Tutorials http://outreach.maizegdb.org	Written and video tutorials that explain how to use various tools on the MaizeGDB website
Data Contribution “How To” Guide http://www.maizegdb.org/contribute_data	Displays sources of currently stored data and how researchers can contribute their own data
Editorial Board http://www.maizegdb.org/hot_new_papers	A list of noteworthy publications selected monthly by the MaizeGDB Editorial Board
Cooperators’ page http://www.maizegdb.org/cooperators	Page of links to resources supporting the cooperative spirit shared among maize researchers
Maize Genetics Cooperation—Newsletter http://www.maizegdb.org/mnl	Makes accessible online copies of the MNL and provides information on how to receive hard copies
Maize Genetics Executive Committee http://www.maizegdb.org/mgec	Provides details on the membership, goals, function, and history of the MGEC
Maize Genetics Conference http://www.maizegdb.org/maize_meeting/	Centralized resource for current and past Maize Genetics Conferences which includes dates, locations, registration information, and abstract submission forms
Maize Research Projects List http://www.maizegdb.org/popcorn/search/	A curated list of maize projects and links to their respective project sites and resources generated
MaizeGDB/National Corn Growers Podcasts (http://www.maizegdb.org/)	An ongoing series of podcasts that educates MaizeGDB’s stakeholders on the value of maize research and the need to make all the data publicly available

1. Sequences from various maize inbred lines, including all versions of the B73 Reference Genome Assembly. Sequences from other inbred lines are shown in relationship to the inbred B73. These are accessible through MaizeGDB’s GBrowse-based Genome Browser.
2. SNPs and flanking genomic sequences from different maize inbred lines, and access to larger databases such as Panzehlha (<http://www.panzea.org/>) and dbSNP (<http://www.ncbi.nlm.nih.gov/SNP/>).
3. BAC sequences from the minimum tiling path of the B73 Reference Genome Assembly, as well as contig information for both within BACs, and between BACs.
4. Molecular probe sequences and detection/amplification methods for RAPDs, ESTs, SSRs, RFLPs, AFLPs, overgos, and other genomic DNAs.

2.2 Large Datasets

With Next Generation Sequencing technologies, there has been a fundamental change in how maize researchers ask and address biological questions. It is now possible to generate large DNA/RNA sequence-based data sets at costs that were previously unthinkable. Because of this, these large datasets (often termed “Big Data” though, as mentioned above, large datasets are only one aspect of what constitutes Big Data) can contain millions, and even billions of data points. Access to these large data sets is currently problematic. Active research is ongoing in many fields to evolve methods of access and display to handle large datasets. Currently at MaizeGDB, various large datasets can be accessed through portals to other databases.

1. RNAseq primary data is stored at the NCBI Sequence Read Archive (SRA, <http://www.ncbi.nlm.nih.gov/sra>), and served at MaizeGDB primarily as graphs of expression at each locus on the Genome Browser.
2. Access to billions of SNPs and flanking sequences from different maize inbred lines is currently available through Panzea and will eventually be accessible directly from MaizeGDB.
3. Proteomic data on the developing maize kernel made accessible through the Maize Proteome Project (<http://maizeproteome.ucsd.edu/>).

2.3 Genetic Data

1. Loci including (but not limited to) genes, chromosomal segments, centromeres, introns, probed sites, and quantitative trait loci (QTL).
2. Variations including known mutant and non-mutant alleles at a given locus, chromosomal structural variations, cytoplasmic variations, DNA polymorphisms, rearrangements, transpositions, etc.
3. QTL experiment environmental conditions, parental stocks, agronomic traits of interest, locus summaries, and raw data files.
4. High resolution genetic maps including nested association maps, hapmaps, and intermated population maps.
5. Over 2000 genetic maps, including both composite and individual maps, cytogenetic and cytological maps along with associated data including mapping panel descriptors, population size, and source information.
6. Seed stock (accession) descriptors consisting of a unique identifier (the stock name) and known synonyms, the stock source (e.g., an individual researcher’s name or an organization name like the “Maize Genetics Cooperation—Stock Center”), and associated locus linkage group assignments, genotypic variations, karyotypic variations, phenotypes, and parental stock identifiers.

2.4 Gene Product and Functional Characterization Descriptions

1. Metabolic pathway data can be accessed through CornCyc, (<http://corncyc.maizegdb.org>, developed in collaboration with the Plant Metabolic Network, PMN) and MaizeCyc (<http://maizecyc.maizegdb.org>, developed in collaboration with Gramene).
2. Gene products with associated Enzyme Commission (EC) numbers, expression, induction conditions, subcellular localization data, metabolic pathway, known metabolic cofactors, mass (kDa), and links to loci that encode them.
3. Phenotypic descriptions that include trait descriptions and affected tissue types/organs (body parts) alongside mutant images.

2.5 Terms, Controlled Vocabularies, and Ontologies

1. Ontologies are hierarchically related controlled vocabularies that serve to enable communication across different databases and data sets. Ontology terms from many different established ontologies, such as the Gene Ontologies [9], the Plant Ontology, and the Trait Ontology [10], are assigned to data as appropriate. Where appropriate, phenotypes are described using Entity-Quality (EQ) statements which utilize the strengths of many different ontologies.
2. Terms and term definitions that describe stored data of various types.
3. Additional controlled vocabularies that are the set of terms that describe a given process or datatype. For example, terms of type “Developmental Stage” make up one controlled vocabulary.

2.6 Literature References and Person/Organization Records

1. References from primary literature, the Maize Genetics Cooperation—Newsletter, and abstracts from the Annual Maize Genetics Conference; associated with virtually all other data types.
2. Contact information records for cooperators, authors, and organizations.

3 Methods

This section outlines the various ways to find and interact with data at MaizeGDB.

3.1 Interrogation Tools

Navigating data to find specific, useful pieces of information is not always a simple task. Learning to use the tools that will enable facile data navigation is, therefore, a good use of time. By learning the general methods for browsing and searching MaizeGDB, the time required to locate information will be decreased, allowing for more to be spent testing hypotheses at the bench. In each of the

following sections, descriptions for each type of search are provided and general techniques for efficiently and effectively navigating the MaizeGDB interface are described.

3.1.1 Interrogation Tools: Embedded Simple Search

The fastest and easiest way to navigate to data of interest at MaizeGDB is by using the search feature located on the right side of the horizontal green toolbar across every page. Clicking on “Search” will open the simple search box where you can enter your search term. If “all data” remains selected in the drop down menu, virtually all data will be searched simultaneously. Much faster searches can be done by selecting the data type from the dropdown menu. Press the button marked “Go!”, or the enter or return button on your keyboard to start the search. Below, instructions are given to find the genetic position of the *bronze1* (*bz1*) locus, its gene model and how to visualize the gene model on the genome browser.

1. Go to [http:// www.maizegdb.org](http://www.maizegdb.org).
2. Locate the green menu bar at the top of the page, and click “search” on the right hand side.
3. Read the note that appears in the popup window.
4. Specify criteria to locate records about *bronze1* by selecting “locus/loci” from the dropdown menu and by typing *bz1* into the field to the right. Click the button marked “Go!”.
5. The locus page for *bz1* gene is first in the results list. Click the link to the *bz1* gene.
6. If a gene model has been linked to the *bz1* locus, it will be present next to the ear of corn. Some gene models have yet to be linked to loci. On this page, scroll down to see the list of BACs, overgos, and other probes known to mark the *bz1* locus.
7. To see the *bz1* gene model on the Genome Browser, look in the “Overview” box. Scroll down to “Associated Gene Models”, and click the “Genome Browser” link.
8. To download sequence, the GBrowse2 tools can be used: in the search box at the top of the view, select “download decorated FASTA file” or “Download Sequence File” from the pull down menu on the left side.

3.1.2 Interrogation Tools: Advanced Search of Data Centers

Questions asked by biologists are complex, so tools that query the database must enable complex queries to be made. MaizeGDB has grouped like data types into “data centers” (e.g., Genes/Gene models, Alleles and Polymorphisms, Expression, Images, Maps, Phenotypes, Sequences, References). This allows researchers to use custom queries that more efficiently search specific data types. For example, the search algorithms are different when searching for a reference, versus a sequence, versus a phenotype, so the advanced

search boxes allow different inputs. These custom searches also allow more logical display of the results.

Each of the Data Centers can be selected from the “Data Center” drop down menu on the green horizontal bar at the top of most pages. Often used Data Centers are also shown as a button in the center of the home page. Data Center names are linked to a page that explains the Data Center and makes available a Simple Search (similar in function to the search available through the search bar described above), an Advanced Search (if relevant, discussed more fully in this section), and a Discussion of the Data Type (written at a level comprehensible by the general public). The Expression Data Center is an exception: currently, this Data Center brings together many offsite tools to access and analysis expression data from various plant sites.

To demonstrate the use and functionality of the various Data Centers’ Advanced Search tools, here are two examples of their use from two disparate Data Centers.

The Gene/Gene Models Data Center. In this Data Center’s Advance search box, you can search for genes by name, type, known phenotypes, gene products, by chromosome or by a combination of these parameters. Below the Advanced search box is a box where you can search for genes and gene models by sequence using BLAST. Below that is a very useful box that allows users to download all gene between two genome coordinates, between coordinates of two markers or BACs, or download all genes on individual BACs. These search features are useful for researchers performing map based cloning. Next is a search box where you can enter a list of gene model names or transcript IDs (up to 8000), and retrieve their sequence. Lastly on this Data Center page are lists of useful links for accessing and downloading more gene and gene model information.

The Maps Data Center. There are over 2000 genetic maps of maize, and finding an appropriate map can be challenging for researchers. At the top of the Maps Data Center is a “Handy Reference” that describes the most commonly used genetic maps, and how they were constructed. On the Maps Data Center, you can search for genetic maps that contain markers of interest, that are from particular sources (individuals, companies and public institutions) or by mapping panel, or by a combination of these variables. Once a map has been chosen, information about that map can be found on its page. Below the advanced Search box, are links to the most commonly used genetic maps.

3.1.3 Interrogation Tools: Finding Projects and Resources

The *PrOject Portal for corn* (POPcorn) was developed as a single entry point for researchers to explore maize projects and resources that have been developed by maize researchers worldwide to advance maize research [11]. Currently, POPcorn contains 159 projects and 137 resources. Projects and resources are distinguished

from each other in that projects are generally knowledge driven with distinct deliverables and a specific endpoint. In contrast, resources, which often extend beyond the lifetime of the project, provide either biological stocks (either DNA based or seed) or software tools for navigating the data that has been developed by POPcorn projects. The central idea driving the development of a single access point for maize projects and resources was that not only would this save maize researchers time and effort in locating projects that might otherwise be overlooked, but also to make sure that when the funding period ended, that valuable data and tools are maintained long term. Currently, POPcorn serves the maize community in three capacities: (1) POPcorn allows localization and utilization of community databases and large scale data sets with a single search. Curated POPcorn projects and resources can be searched by keyword, investigator, institution, category, and country. (2) POPcorn allows single or multiple DNA sequence searches (via BLAST) that access all POPcorn associated DNA sequence databases (~45 are currently represented) and returns a single, collated output for easy viewing. (3) POPcorn provides a mechanism for preserving raw data and associated annotations contained within POPcorn for migration to MaizeGDB for long-term storage. An important feature of POPcorn is that it is still actively curated even though project funding (NSF DBI 074804) has ended. MaizeGDB curators spend a few hours each month looking for new projects and resources while an automated utility checks URLs and send an email to alert curators that a particular link has become inactive. These curation efforts ensure that POPcorn will continue to be a relevant resource to the maize community.

3.1.4 Interrogation Tools: Accessing the Maize Community

The maize community spans a timeline of over 80 years and has a rich tradition of sharing community resources, both physical and intellectual. Not surprisingly, maize as both an applied and model research organism has benefitted from the cohesiveness of the maize community. One of MaizeGDB's primary objectives is to serve as a clearing-house for organizational information of interest to the maize community. Maize researchers can access the MaizeGDB community page from the menu bar on the home page. Contact information for maize researchers (Cooperators) as well as the Maize Executive Committee can be located. Information on the Annual Maize Genetics Conference, including direct links to registration and hotel accommodation's can be obtained. In addition, researchers can get directions and guidelines on contributing their data to MaizeGDB or becoming a community data curator. Information on the Maize Genetics Newsletter, current job listings, and the Maize Editorial Board's recommended readings are also available.

3.2 Analysis Tools

MaizeGDB provided open source and custom tools to allow users to drill down to the data the way they need to view it and use it.

3.2.1 Analysis Tools: BLAST

MaizeGDB created a powerful customizable BLAST tool which can be easily accessed from the home page through the BLAST pull down menu, by selecting BLAST in the Tools pull down menu, or by using the BLAST button in the center of the page. Using webservices, MaizeGDB BLAST can search locally stored sequences, all GenBank datasets (ESTs, GSS, HTGS, etc.), and many other sequence databases such as all gene model builds, all sequence assemblies, repeat databases, all known loci, microarray probes, transcription factors, and more. Selection of datasets to query is at the users' discretion. MaizeGDB BLAST will accept up to five input queries with a combined length of no more than 35,000 bp. Some web services may reject queries of this maximum size; if that happens the results page will show an error for that target host/data set. Longer query sequences or a large number of sequences can be aligned to the reference sequence assembly using another tool called [ZeAlign](#) (described below). MaizeGDB BLAST has four clearly delimited steps: Step 1: Input your sequences (Raw, FASTA, or GenBank IDs), and indicate the sequence is nucleotides or amino acids. Step 2: Select Datasets: MaizeGDB offers a wide variety of sequence datasets to search. This allows users to have a single place to BLAST where they can hit multiple databases at once. Each dataset is explained by hovering over the title of the dataset. Step 3: Select BLAST parameters: by choosing one of the preset options, or by modifying the advanced settings. An explanation for each option is available by hovering over the option title. Step 4: Select output type: MaizeGDB offers three possible outputs layouts; The standard BLAST text output (like the BLAST output at NCBI), The BLAST table output, and a expanded table output created by MaizeGDB for maximum integration with the website. Results can be displayed, or emailed, or both.

3.2.2 Analysis Tools: ZeAlign

Users that generate a large number of sequences can use the MaizeGDB developed ZeAlign tool to align their sequences to the current genome Assembly. This tool can also be used to remap older data to a newer genome assembly. ZeAlign is a BLAST tool, but unlike the regular MaizeGDB BLAST tool, ZeAlign allows up to 20,000 sequences to be aligned to a genome assembly. This tool is often used before researchers put their data on their own custom genome browser track which can be either private, or can be submitted to MaizeGDB for review and if approved, then for public viewing. ZeAlign can be reached by selecting it from the BLAST pull down menu on the green bar across the top of every page.

3.2.3 Analysis Tools: Genome Browser

MaizeGDB users can access five genome browsers covering three genotypes from the MaizeGDB home page: (1) MaizeGDB/B73, (2) Maizesequence.org/B73, (3) Genomaize/B73, (4) Phytozome/Mo17, (5) and QuerySequenceVisualizer/Palomero Toluqueño. The MaizeGDB Genome Browser uses a semi-customized version of GBrowse2 and is actively supported by MaizeGDB personnel.

The MaizeGDB genome browser allows access to archived sequence assembly versions (such as the original BAC-based sequence of B73 and B73 RefGen_V1), and fully supports the two most recent versions which currently are the B73_RefGen_V2 and B73 RefGen_V3 maize genome assemblies. This is critical to many MaizeGDB users as researchers often become vested in a particular version of the genome assembly. Genome version stability is essential for them to complete their research projects—especially those taking a positional cloning approach towards gene cloning. The current installation of GBrowse2 at MaizeGDB contains over 50 tracks that provide information on genome diversity, gene expression, gene models, genetic maps, DNA insertions, and repetitive elements. In addition to the standard GBrowse2 features such as the option to create snapshots and generate community and/or private tracks, MaizeGDB has created maize-specific documentation and tools tabs. The documentation tab introduces new users to various features in GBrowse2, whereas the maize specific tools tab gives users access to maize specific tools, such as BLAST, chromosomal bin, and incongruence viewers that have been embedded within the GBrowse2 environment.

3.2.4 Analysis Tools: *Locus Lookup and Locus Pair Lookup*

Not all genes in the maize genome have been identified, and not all in silico gene models have been linked with genes previously identified by classical genetics. Often researchers identify new genes based on mutant phenotype, genetically map them, and then want to identify their sequence. MaizeGDB provides the Locus Lookup and Locus Pair Lookup tools [12] to locate a genomic sequence interval where a gene of interest may reside, based on its position on the genetic map. This tool can be used if you have a genetic position for your gene of interest relative to one of several MGDB's composite genetic maps. The tool works by finding the nearest genetically mapped loci that flank the input locus and have an association with the B73 genome sequence, and returns those coordinates. Since many classically identified genes do have sequence coordinates, the Locus lookup tool goes through a hierarchical process:

Upon given the input coordinates, the tool:

1. checks if the locus is associated to any gene models and the coordinates for the gene model are returned, else
2. checks physical map coordinates to find out whether the locus is already placed. If so, the physically mapped locus coordinates are returned, else
3. checks the locus record at MaizeGDB to find out if any placed BACs are known to detect the locus and that BAC is returned within its genomic context, else
4. genetically mapped probes that are nearest the input locus are identified, the tool checks whether those probes have known

genomic coordinates (working outward until appropriate probes are identified) and finally the region of the genome contained by the identified probes is reported with bounding probes shown in red.

The Locus Lookup tool will locate the genome region around one genetically mapped locus, while the Locus Pair Lookup allows the user to enter two genetically mapped loci and returns the genome sequence coordinates that include both loci and the region between them. The Locus lookup tool can be accessed from the home page by clicking the large "Locus Lookup" button, or by selecting this under the "Tools" drop down menu.

3.2.5 *Analysis Tools: Cyc Databases*

Metabolic pathways hosted at MaizeGDB can be accessed through either the expression or metabolic pathways quick links on the MaizeGDB homepage. Currently MaizeGDB hosts two independent, but complementary metabolic pathway viewers, MaizeCyc developed by Gramene [13, 14], and CornCyc, created by the Plant Metabolic Network which were both computationally inferred using either the Pathway Tools Software suite (MaizeCyc) or the Ensemble Enzyme Prediction Pipeline (E2P2. CornCyc). For MaizeCyc, MaizeGDB curators provided 772 literature-based GO term annotations, while for CornCyc, they provided curations on the auxin, brassinosteroid, and gibberellin pathways. The MaizeCyc and CornCyc respective pipelines identified similar numbers of enzymatic reactions and pathways. However, the two differ in that MaizeCyc contains a larger numbers of transporter proteins and reactions while CornCyc identifies over 4000 spliced protein variants. MaizeCyc and CornCyc were developed using different stringencies for their respective pipeline so it is not surprising their output differs with respect to the number of protein functions identified. Taken together, these two approaches are highly complementary and provide a more robust resource than either would alone: MaizeCyc identifies more enzymes and pathways but at lower accuracy, whereas CornCyc identifies fewer, but at a higher accuracy.

3.2.6 *Analysis Tools: Expression Data Center*

The expression data center at MaizeGDB houses both gene expression data as well as a collection of tools designed to facilitate its analysis. It can be reached either through the expression quick link on the MaizeGDB home page or from the data centers pull down menu at the top of the home page. The primary focus of the expression data center is to provide MaizeGDB users with access to a suite of complementary utilities that have been developed by external groups to leverage atlas style, genome-wide, gene expression data sets that have been generated by either high density microarrays or RNA sequencing. The gene expression tools accessible from MaizeGDB accommodate virtually any approach that a researcher might take towards leveraging expression data. The

maize eFP browser, developed by researchers at the University of Toronto [15] uses a pictogram approach to display the level of gene expression across a series of maize pictures representing over 60 maize tissues or cell types. It also allows the gene expression patterns of genes to be compared across tissues. This is particularly useful in comparing closely related homologs and identifying tissue specific expression patterns. The MapMan utility developed at the Max Planck Institute for Molecular Plant Physiology projects gene expression data on to a large collection of biochemical processes and metabolic pathways [16]. Users can compare and contrast two tissues or treatments across a range of metabolic pathways and quickly identify gene expression differences within specific steps in a biochemical pathway. MapMan is particularly well suited for control vs. experimental treatments such as mutant vs wild type or abiotic stress treatments where major shifts in metabolic pathways might be expected. The qTeller (QTL Teller) utility, which was developed by James Schnable and Mike Freeling at the University of California-Berkeley, allows users to view expression levels of genes within a user specified chromosomal interval. qTeller, which draws upon a large corpus of publically available gene expression data, also allows the expression levels of syntenic orthologs of rice, sorghum, *Setaria*, and *Brachypodium* to be compared which facilitates cross species comparisons of genes within a syntenic interval. Due to these unique features of qTeller, it is the expression analysis tool of choice for positional or QTL cloning projects.

3.2.7 Analysis Tools: *Incongruency Tool*

Maize has a large genome (about 2.7 gigabases [17]), many diverse lines with different DNA content, and the B73 reference sequence assembly is in flux. The B73 inbred line of maize was sequenced in a BAC by BAC approach (pubmed/19965430). First, a BAC library was made, then the identification of genetically mapped probes on each BAC allowed BACs to be aligned to the IBM genetic map, creating a high information content fingerprint (HICF) physical map. From this, a minimum tilling path of BACs was selected, and these BACs were sequenced, starting in 2005. Thus, the order of the sequences of each BAC is determined by the IBM genetic map. Since 2005, the density of markers per BAC (i.e., the complete sequence of the BAC) has increased, and the IBM genetic map has improved. The Incongruency tool allows users to check genomic regions for inconsistencies between the current genome assembly and the current IBM map (ISU Integrated IBM 2009 map). Where there is a discrepancy, it is difficult to determine which version is correct, however, the tool indicates a region of the genome assembly where the assembly itself needs improvement and serves as a warning to researchers to be careful with interpreting data in these regions. The Incongruency tool can be accessed from any page using in the TOOLS drop down menu.

3.2.8 Analysis Tools: Bin Viewer

Sometimes users want to see all the data available in a small region of the genome. While the Genome Browser is an excellent way to get data in a known sequence region, the Bin Viewer allows users to get data in a defined genetic region. Each maize chromosome is divided into 7–12 regions that are each about 20 centiMorgans [18]. To view all data in each bin, users can get to the Bin Viewer from the home page by clicking the centrally located Bin Viewer button, or selecting it from the “Tools” drop down menu. The Bin Viewer is a particular good way to visualize QTL data.

3.3 Ways to Add Data to the Database

MaizeGDB pulls sequence and other data from many sources. Other datasets are classified as one of three types: large datasets, small datasets, and notes. Large datasets are generally added to the database in bulk by members of the MaizeGDB Team, and are contributed by researchers directly. To contribute a large dataset to the project (or to find out whether the dataset you have generated constitutes a “large” or “small” dataset), use the feedback button at the top of any MaizeGDB page to make an inquiry.

Researchers can add “notes” to records. To add such a note, log in to the site using the “annotation” link displayed at the top right of any MaizeGDB page. Once logged in, click the “Add your own annotation to this record” link shown at the top of virtually all data displays. You may also add GO annotation in this way. Small datasets can be added to the database by researchers directly by way of the MaizeGDB Community Curation Tools. The method for adding a small dataset is explained below, using a newly published reference as the example usage case. The citation for our pretend reference is as follows:

Lawrence, CJ. (2005) How to use the reference curation module at MaizeGDB. *Plant Physiology* 9:3–4.

1. Click on the “annotation” link at the top of any MaizeGDB page. Click the link to “Create an Annotation Account” and fill out all information required. Be sure to check the box to become a MaizeGDB curator before clicking the submit button.
2. A confirmation email along with a Community Curation manual will be sent once the new account has been activated.
3. To begin adding data to the database, click on the link marked “tools” toward the top right of any MaizeGDB page.
4. Toward the bottom of this page click the link marked “Playground Community Curation Tools”.
5. Log in using the newly created username and password.
6. Click the link toward the center of the page to download the curation tools’ user manual for future reference.
7. In the left bar, click the link marked “Reference.”
8. Fill in the title and select “article” as the reference type. (Because you are working at the “Playground Community

Curation Tools” feel free to make up pretend information for the purposes of this exercise.) When in doubt of what information to put into a given field, click on the buttons labeled with a question mark.

9. Fill in the year, volume, and pages information. For the “In Journal” field note the label “Lookup Field—Enter a Search String.” Fill in the journal title.
10. Click the link beneath the “Author” heading to “Add Authors.”
11. Half way down the page is a text field where the name of an author can be typed to locate a person record to associate with the new reference. For this example, type *Lawrence*.
12. Lawrence is the first (and only) author on this imaginary publication, so leave the dropdown menu with “Author” selected, and type the number **1** into the box labeled “Order.” Press the “Submit & Continue” button.
13. Note that “Lawrence, CJ” is available in the dropdown menu. Select this item from the dropdown menu that has replaced the typing field for Author, scroll to the bottom of the page, and click the button labeled “Add to List of Authors.”
14. Note that “Lawrence, CJ” now appears in the list of authors at the top of the page. Click the button marked “Author List Complete.”
15. Scroll to the bottom of the page and press the button marked “Submit & Continue.” Returned in place of the “In Journal” search string are available instances of matching journal names. Select the records “Plant Physiol”.
16. Click the button at the bottom of the page marked “Insert into Database.”

The newly created record enters a queue for approval by a worker at MaizeGDB. Once the record has been approved, it will become available through the MaizeGDB interface after the next database update. Other curation tool modules function similarly, and a detailed manual is available through the curation tools.

3.4 Leave Feedback

Personnel at MaizeGDB aim to be responsive to the questions and requests of people who use this resource. At the top of every page, on the horizontal green bar, is a feedback button. When used, it will record the page from which the feedback was initiated. Users can ask questions, make comments, and point out errors. MaizeGDB strongly encourages everyone to use the feedback button often!

3.5 Outreach

MaizeGDB provides written, video, and in-person tutorials on a broad range of topics all relating to interacting effectively with data at MaizeGDB [19, 20]. All tutorial and FAQ information can be found in the outreach section on the bottom left of the home page, Links to the MaizeGDB Facebook and Twitter pages are also there.

Acknowledgments

The author would like to thank Darwin Campbell for his work in maintaining the MaizeGDB database; Mary Schaeffer, Ed Coe, and Marty Sachs for their curatorial work; and Ethalinda Cannon, Bremen Braun, Scott Burket, John Portwood, and Jackie Richter for their technical support. This work was supported by the USDA-ARS, and the National Corn Growers Association.

References

1. Lawrence CJ, Dong Q, Polacco ML, Seigfried TE, Brendel V (2004) MaizeGDB, the community database for maize genetics and genomics. *Nucleic Acids Res* 32:D393–D397
2. Polacco M, Coe E (1999) In: Letovsky SI (ed) *Bioinformatics: databases and systems*. Kluwer Academic Publishers, Norwell, MA, pp 151–162
3. Polacco M, Coe E, Fang Z, Hancock D, Sanchez-Villeda H, Schroeder S (2002) MaizeDB—a functional genomics perspective. *Comp Funct Genomics* 3(2):128–131
4. Dong Q, Roy L, Freeling M, Walbot V, Brendel V (2003) ZmDB, an integrated database for maize genome research. *Nucleic Acids Res* 31:244–247
5. Sen TZ, Andorf CM, Schaeffer ML, Harper LC, Sparks ME, Duvick J, Brendel VP, Cannon E, Campbell DA, Lawrence CJ (2009) MaizeGDB becomes ‘sequence-centric’. Database. doi:10.1093/database/bap020
6. Lawrence CJ, Harper LC, Schaeffer ML, Sen TZ, Seigfried TE, Campbell DA (2008) MaizeGDB: the maize model organism database for basic, translational, and applied research. *Int J Plant Genomics*. doi:10.1155/2008/496957
7. Lawrence CJ, Schaeffer ML, Seigfried TE, Campbell DA, Harper LC (2007) MaizeGDB’s new data types, resources and activities. *Nucleic Acids Res* 35(Database issue):D895–D900
8. Lawrence CJ, Seigfried TE, Brendel V (2005) The maize genetics and genomics database: the community resource for access to diverse maize data. *Plant Physiol* 138:55–58
9. Gene Ontology Consortium (2001) Creating the gene ontology resource: design and implementation. *Genome Res* 11:1425–1433
10. Bruskiwich R, Coe E, Jaiswal P, McCouch S, Polacco M, Stein L, Vincent L, Ware D (2002) The plant ontology consortium and plant ontologies. *Comp Funct Genomics* 3:137–142
11. Cannon EK, Birkett SM, Braun BL, Kodavali S, Jennewein DM, Yilmaz A, Antonescu C, Schaeffer ML, Campbell DA, Andorf CM, Andorf D, Lisch D, Koch KE, McCarty DR, Quackenbush J, Grotewold E, Lushbough CM, Sen TZ, Lawrence CJ (2011) POPcorn: an online resource providing access to distributed and diverse maize project data. *Int J Plant Genomics*. doi:10.1155/2011/923035
12. Andorf CM, Lawrence CJ, Harper LC, Schaeffer ML, Campbell DA, Sen TZ (2010) The Locus Lookup tool at MaizeGDB: identification of genomic regions in maize by integrating sequence information with physical and genetic maps. *Bioinformatics*. doi:10.1093/bioinformatics/btp556
13. Monaco MK, Senbc TZ, Dharmawardhanad PD, Rena L, Schaefferbe M, Naithanif S, Amarasinghed V, Thomasona J, Harperbg L, Gardinerch J, Cannonc EKS, Lawrencebc CJ, Wareab D, Jaiswal P (2013) Maize metabolic network construction and transcriptome analysis. *Plant Genome*. doi:10.3835/plantgenome2012.09.0025
14. Monaco MK, Stein J, Naithani S, Wei S, Dharmawardhana P, Kumari S, Amarasinghe V, Youens-Clark K, Thomason J, Preece J, Pasternak S, Olson A, Jiao Y, Lu Z, Bolser D, Kerhornou A, Staines D, Walts B, Wu G, D’Eustachio P, Haw R, Croft D, Kersey PJ, Stein L, Jaiswal P, Ware D (2013) Gramene 2013: comparative plant genomics resources. *Nucleic Acids Res*. doi:10.1093/nar/gkt1110
15. Winter D, Vinegar B, Nahal H, Ammar R, Wilson GV, Provart NJ (2007) An “electronic fluorescent pictograph” browser for exploring and analyzing large-scale biological data sets. *PLoS One* 2(8), e718
16. Thimm O, Blasing O, Gibon Y, Nagel A, Meyer S, Krüger P, Selbig J, Müller LA, Rhee SY, Stitt M (2004) MAPMAN: a user-driven tool to display genomics data sets onto diagrams of metabolic pathways and other biological processes. *Plant J* 37(6):914–939

17. Laurie DA, Bennett MD (1985) Nuclear DNA content in the genera *Zea* and *Sorghum*—intergenic, interspecific and intraspecific variation. *Heredity* 55:307–313
18. Gardiner JM, Coe EH, Melia-Hancock S, Hoisington DA, Chao S (1993) Development of a core RFLP map in maize using an immortalized F2 population. *Genetics* 134(3):917–930
19. Schaeffer ML, Harper LC, Gardiner JM, Andorf CM, Campbell DA, Cannon EK, Sen TZ, Lawrence CJ (2011) MaizeGDB: curation and outreach go hand-in-hand. *Database* (Oxford). doi:[10.1093/database/bar022](https://doi.org/10.1093/database/bar022)
20. Harper LC, Schaeffer ML, Thistle J, Gardiner JM, Andorf CM, Campbell DA, Cannon EK, Braun BL, Birkett SM, Lawrence CJ, Sen TZ (2011) The MaizeGDB Genome Browser tutorial: one example of database outreach to biologists via video. *Database* (Oxford). doi:[10.1093/database/bar016](https://doi.org/10.1093/database/bar016)

Chapter 10

WheatGenome.info: A Resource for Wheat Genomics Resource

Kaitao Lai

Abstract

An integrated database with a variety of Web-based systems named WheatGenome.info hosting wheat genome and genomic data has been developed to support wheat research and crop improvement. The resource includes multiple Web-based applications, which are implemented as a variety of Web-based systems. These include a GBrowse2-based wheat genome viewer with BLAST search portal, TAGdb for searching wheat second generation genome sequence data, wheat autoSNPdb, links to wheat genetic maps using CMap and CMap3D, and a wheat genome Wiki to allow interaction between diverse wheat genome sequencing activities. This portal provides links to a variety of wheat genome resources hosted at other research organizations. This integrated database aims to accelerate wheat genome research and is freely accessible via the web interface at <http://www.wheatgenome.info/>.

Key words *Triticum aestivum*, Integrated database, Comparative map, Geneticmap, Genomeviewer, Second-generation DNA sequencing

1 Introduction

Next-generation high-throughput DNA sequencing (NGS) technologies, also known as second-generation sequencing (SGS) technologies, have provided fascinating opportunities for the analysis of plants on a genomic scale [1, 2]. Wheat is an example of a major polyploid crop with large genome and high complexity presenting significant challenges for analysis [3]. Bread wheat (*Triticum aestivum*) is an important crop plant and staple food worldwide [4]. Here I describe public resources for the study of the emerging wheat genome.

2 Materials and Methods

2.1 *Wheat Group 7 Sequence and 4AL*

Despite the significance of wheat for food globally, its complex and large genome impedes efforts in genome sequencing. Berkman et al. have assembled genomic regions representing unique and low copy regions for isolated chromosome arms. These genomic regions include syntenic builds for chromosomes 7A, 7B, and 7D [5–7]. The syntenic builds represent gene containing contigs which demonstrate similarity with genes from related species, and represent around 4 % of the total assembly. These genomic regions, containing all or nearly all genes for these chromosomes, have been assembled, and the majority of these genes have been ordered and aligned based on synteny with *B. distachyon*, *O. sativa*, and *S. bicolor* [7]. The majority of wheat contigs, which are outside of the syntenic builds, are also included in the WheatGenome.info database.

2.2 *Brachypodium distachyon Gene and Exon Annotation*

Grasses not only provide the bulk of nutrition for humans, but also produce a source of sustainable energy [8]. *Brachypodium* is a member of the Pooideae subfamily. The diploid ecotype of *Brachypodium distachyon* has the smallest reported genome size in the Poaceae [9] and this was the first member of the Pooideae subfamily to be sequenced (published by The International Brachypodium Initiative). *B. distachyon* is an important model for developing new energy and food crops [10].

2.3 *Uniref90 Annotation*

Clustering of protein sequence space based on sequence similarity provides information for reducing overrepresentation of sequences. The UniRef (UniProt Reference Clusters) provide clustered sets of sequences from UniProt Knowledgebase (UniProtKB) and selected UniProt Archive records. UniRef90 are constructed by clustering UniRef100 sequences at the 90 % sequence identity level [11]. The UniRef90 database was downloaded from the UniProt website, and the wheat chromosomes and extra contigs were compared with the UniRef90 database using BLAST. Uniref90 protein annotation can provide biological information related to specific genes or traits.

2.4 *Intervarietal SNPs Between 16 Australian Wheat Varieties*

To obtain a greater understanding of wheat genome diversity, we have identified intervarietal single nucleotide polymorphisms (SNPs) between 16 Australian bread wheat varieties, which include AC Barrie, Alsen, Baxter, Chara, Drysdale, Excalibur, Gladius, H45, Kukri, Pastor, RAC875, VolcaniDDI, Westonia, Wyalkatchem, Xiaoyan 54, and Yitpi. These 16 Australian bread wheat varieties were sequenced using whole genome shotgun Illumina paired read sequencing [12]. Lai et al. have aligned these paired reads to the draft assemblies of chromosomes 7A, 7B, and

7D. A total of over 4,018,311 intervarietal SNPs have been identified between these 16 Australian wheat varieties [13].

3 Methods and Workflow

The WheatGenome.info integrated database and portal can be split into distinct sections: the generative pipeline, the data storage component, and the visual interface (Fig. 1). In pipeline processing, BLAST tools were used for comparison with genome references and annotation. Annotated genetic maps were generated in CMAP. All of the processed data is stored in a MySQL relational database using customized schema for data storage. Users interact with each Web-based system through their respective portals.

3.1 The Entry Page—A Summary and a Gateway

The entry page of wheatgenome.info provides four links. The introduction link provides summary information about the initiative of this wheatgenome.info project. The second link provides the details of this project, including the purpose of this project, the method of whole-genome shotgun (WGS) to help wheat genome sequencing, and the challenges of wheat genome sequencing. The third link provides a summary page of links to different wheat genome references in the GBrowse 2 viewer, and the links to the TAGdb, BLAST portal, and wheat genome CMap (Fig. 2). The fourth link provides resources, and external wheat databases and information from other research teams.

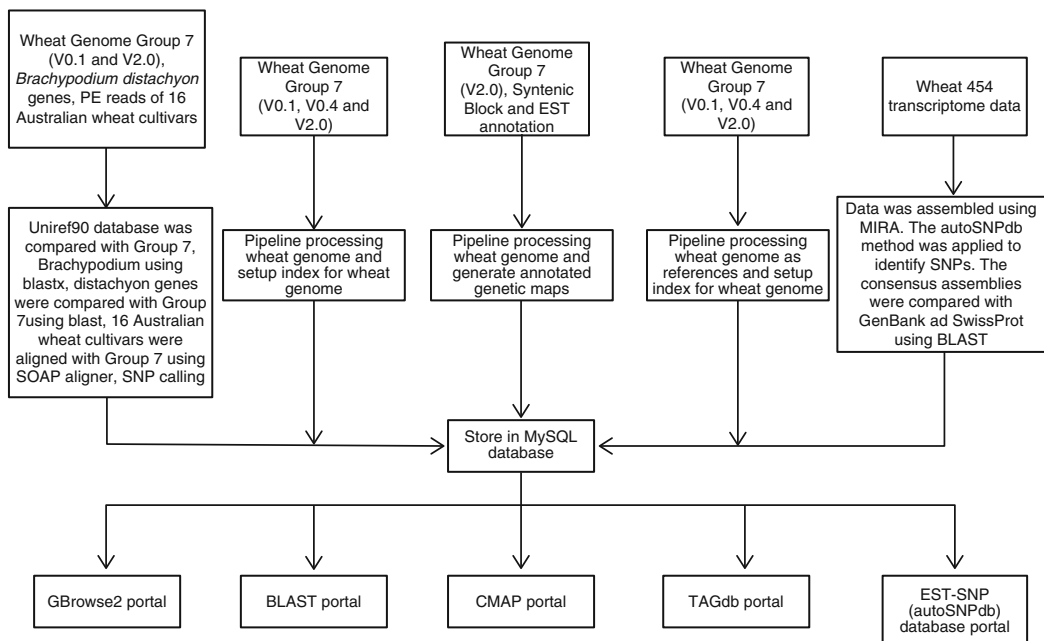


Fig. 1 Flowchart demonstrates the general workflow of wheatgenome.info integrated database

Introduction
Details
Databases
Resources

Wheat Genome

Wheat Genome Databases

Individual chromosome arms are being assembled and annotated. They are made publically available as they are produced using the genome viewer GBrowse2.

Links to available chromosome arms are below.

All chromosome arm specific sequence data is available at [TAGdb](#).

All raw and assembled sequence data is freely available [on request](#).

Wheat GBrowse viewers and search

1. [Wheat 7A v2.0](#)
2. [Wheat 7B v2.0](#)
3. [Wheat 7D v2.0](#)

Wheat GBrowse databases

1. [Wheat 7AS v0.1](#)
2. [Wheat 7AL v0.1](#)
3. [Wheat 7BS v0.1](#)
4. [Wheat 7BL v0.1](#)
5. [Wheat 7DS v0.1](#)
6. [Wheat 7DL v0.1](#)

[Wheat genome assembly BLAST portal](#)

[Wheat genome cmap](#) - comparative genome and genetic maps

Please cite

[WheatGenome.info: An integrated database and portal for wheat genome information](#). Kaitao Lai, Paul J Berkman, Michal Tadeusz Lorenc, Christopher Duran, Lars Smits, Sahana Manoli, Jiri Stiller, David Edwards. *Plant and Cell Physiology* (2012) 53(2): e2.

Others: If we have missed a link to your site, please contact the [web admin](#)

If we have missed a link to your site, please contact the [web admin](#)
The site is supported by funds from the [University of Queensland](#) and the [Australian Research Council](#).

Fig. 2 Wheat GenomeDatabases list

3.2 GBrowse2 Viewer for Wheat Genomes

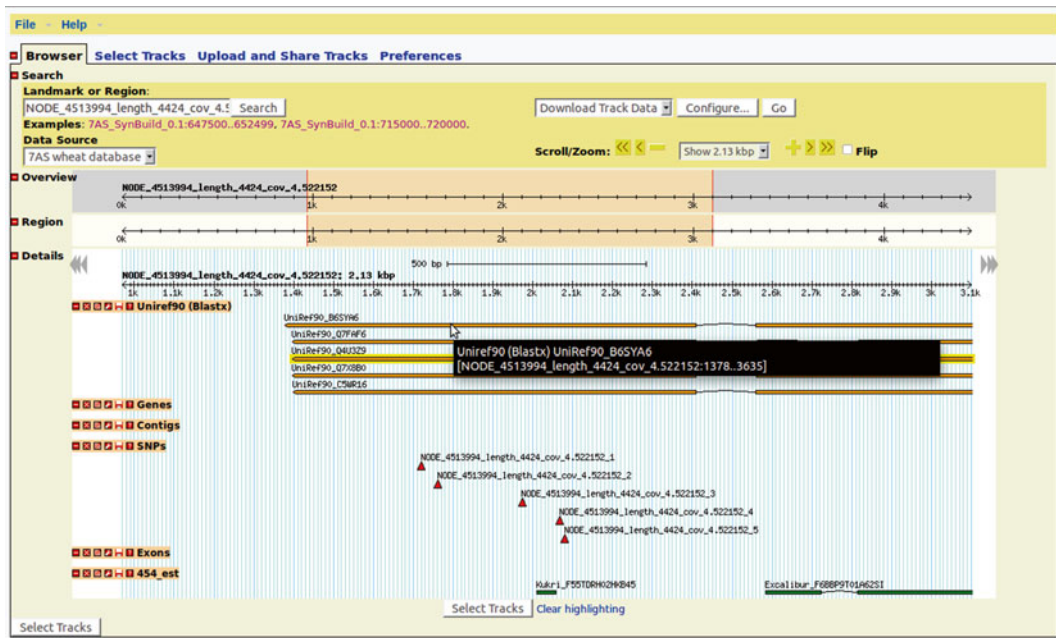
GBrowse2 is a genome browser that allows users to visualize DNA, protein, or other sequence features within the context of a reference sequence, for example, a chromosome or contig [14]. This web portal applies GBrowse2 to host wheat group 7. GBrowse2 is an interactive generic genome browser for genome sequence data and aligned annotation [15, 16]. Predicted genes, Uniref90 genes

[17] as well as intervarietal SNPs between 16 Australian wheat varieties have been annotated on each wheat chromosome arm (syntenic build).

The data file for GBrowse2 is based on a General Feature Format (GFF), which is organized into nine columns, separated by tabs [14]. The user can type in chromosome or contig name (or with length range) in the search box. The results would present the related information within the context of the searched sequence.

The display panel has several sections (Fig. 3). The top section displays the overview of the searched chromosome or contig. Users can drag and select the range of sequence region, the bottom section (details section) presents the DNA regions, proteins, and other sequence features within the selected region. The detail section is a sliding window, which can be dragged and moved left or right to view other regions of the chromosome or contig.

When the user selects a sequence to display, the detail section will display Uniref90 proteins, the assembled contigs, *Brachypodium distachyon* genes and their exons, and the intervarietal SNPs between 16 Australian wheat varieties. All of the sequence features provide links to a detail page. When the user clicks any of these sequence features, the detail page will be displayed with feature



Generic Genome Browser version 2.14. For questions about the data at this site, please contact its webmaster. For support of the browser software only, send email to gmod-gbrowse@lists.sourceforge.net or visit the [GMOD Project](http://gmod.org) web pages.

Fig. 3 Example of the detailed information for the wheat genome 7AS syntenic build from the reference view of GBrowse2. Several tracks of annotation are available, including Uniref90, Genes, Contigs, SNPs, and Exons [17]

name, type name, description, reference position, length, and its full sequence.

Wheat GBrowse2 also provides the facility to “Download Track Data” and a related configuration button. The user can select GFF version for the output file and can get the download the GFF file which includes information within the selected range of chromosome or contig of interest.

3.3 TAGdb for Interrogating Short Read Sequence Data

TAGdb is a Web-based query tool for aligning query sequences to a database of Illumina short read sequence data [18], including data from multiple species, such as Barley, Brassica, and wheat. The wheat paired short read data library includes 16 Australian bread wheat varieties and Chinese Spring. In the short paired-read library selection box, each library is displayed in a fixed format. This format is described as “Source Name – Read Length – Insert Size – Library Name.”

The e TAGdb portal page provides the user Email input text box, and a text box for input query sequences, and a button for selecting a sequence file. The user needs to choose a sequence query file or copy and paste query sequences in FASTA format. In the next step, the user needs to select the sequences database for the aligning query sequence. After choosing the species, the related short paired-read libraries need to be selected. These libraries can be multiple selected. If the user needs to select many paired-read libraries, they are recommended to submit multiple jobs, and for each job only select ten short paired-read libraries or less.

TAGdb works in an automatic workflow. When a job is submitted, the system will automatically send an email to the user with a unique job reference number, and start to align the query sequence with selected short reads using MEGABLAST [19]. This e-mail will provide a link to the job status web page.

Once the job has been completed, the TAGdb system will send another e-mail to inform the user that the job has completed and provide link to the results (Fig. 4).

In the result page, TAGdb presents the alignment of short reads from wheat libraries visually using OpenLayers. The result view has been separate two sections. The top section is a sliding window. User can drag and move this window and view the details of the alignment on the bottom section. Each aligned paired-read from query sequences are represented with single or paired coloured arrows.

3.4 Wheat GenomeAs semblyBLAST GBrowse2 Portal

The wheat BLAST GBrowse2 portal performs BLAST alignment between a single query and the genome reference. This portal provides parameter drop boxes and an input box. The user can select the E-value threshold. Its default value is $1E-4$. Max hits is defined as 15 by default.

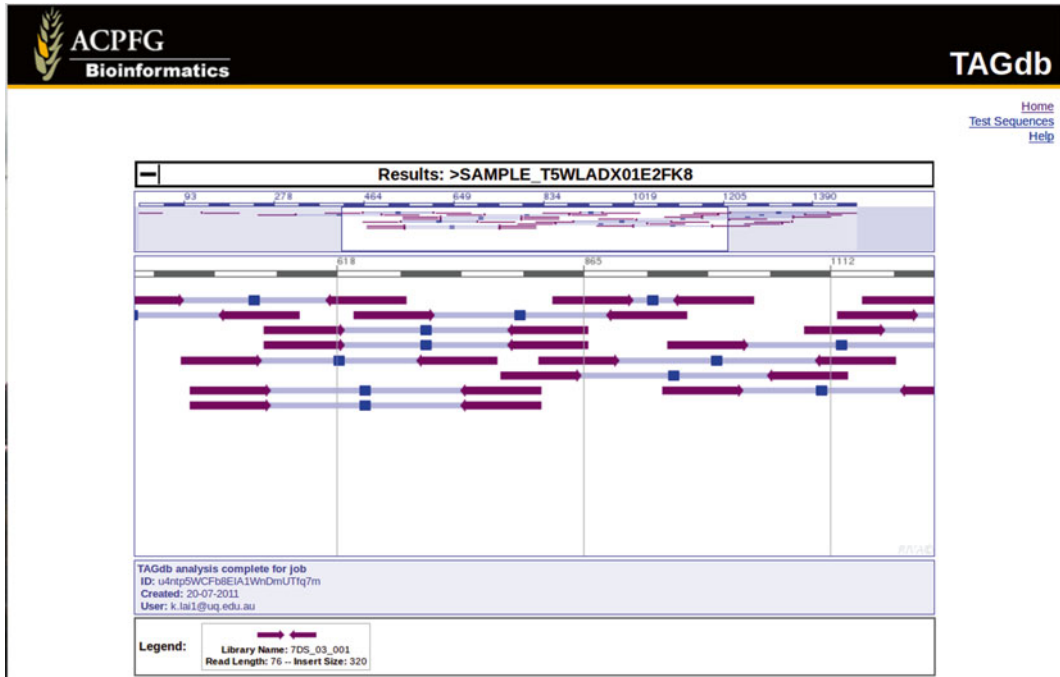


Fig. 4 Screenshot of TAGdb result page demonstrating the alignment of short reads from wheat variety Chinese Spring to a sample query sequence [17]

Sequence data input is similar to TAGdb system. The user needs to select the sequences database for aligning the query sequences. These libraries can be multiple selected. After BLAST alignment has completed, the web portal will present a page to display the standard BLAST result. Each alignment result displays the alignment score and HSP. The user can click score link and the page will forward the alignment details. The user can click the HSP value and the related GBrowse2 view for this range of the genome and sequence features will be displayed.

3.5 Wheat Genome CMap

The CMap system provides a generic, extensible Web-based comparative map viewer for demonstrating and comparing genetic and physical maps from any species [20]. Wheat CMap system has linked the assembled wheat chromosome arm information with the sequenced genomes of *Brachypodium distachyon* and rice, as well as a genetic map of the D genome donor of hexaploid wheat, *Aegilops tauschii* [17].

Wheat CMap provides a summary interface linking to the wheat CMap viewer, administration, tutorial document, map search and feature search functionalities [17]. The user can select the reference species as wheat, and reference physical sequence map or genetic map, and then add a physical sequence map or a

genetic map as second map on the left side or right side of the first map. CMap links the same features between two physical or genetic maps (Fig. 5).

3.6 Wheat Genome CMap3D

The effective visualization of links between genetic and physical maps becomes a challenge when these maps become more abundant [17]. If the links are demonstrated within three maps, the effective visualization would be depreciated. The CMap3D tool is developed based on CMap for the visualization and comparison of multiple physical or genetic maps [21].

In the CMap3D viewer (Fig. 6), the multiple physical or genetic maps are displayed in virtual space. The view of links between any two of these maps is also displayed.

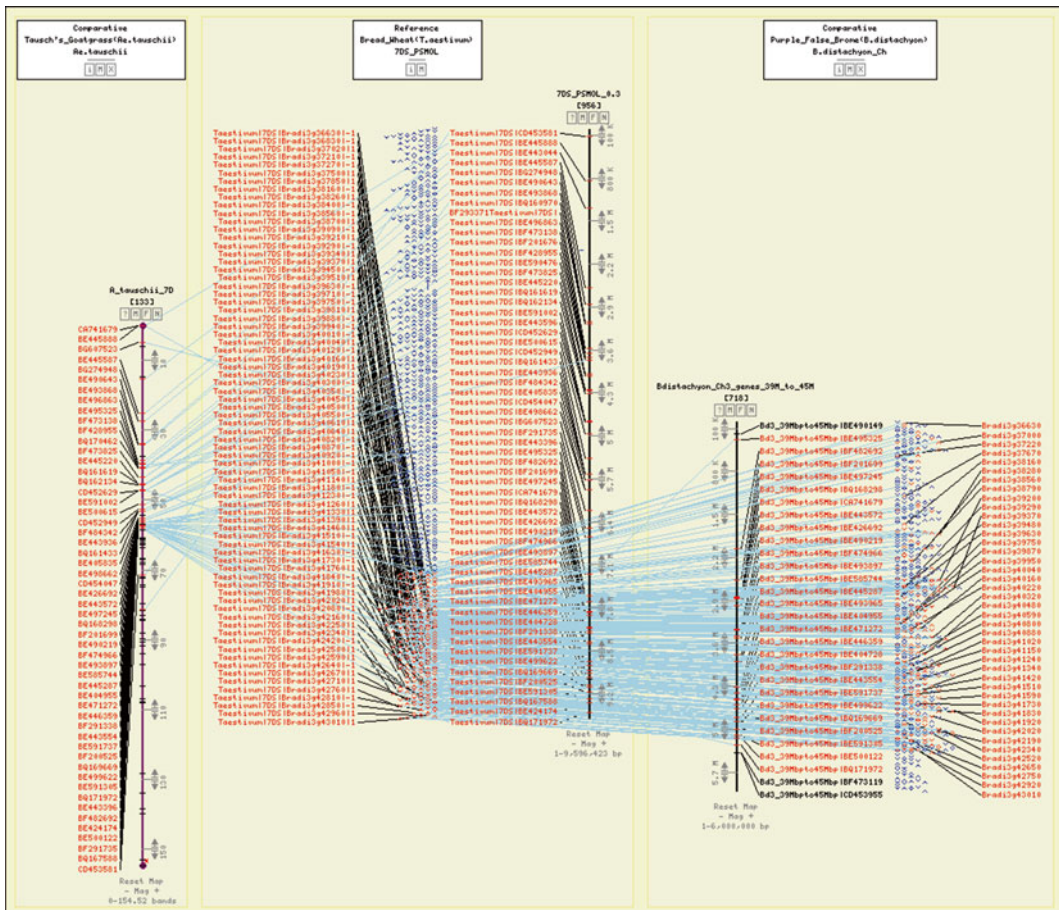


Fig. 5 An inter-species comparison between a physical map of the wheat 7DS syntenic build chromosome, a genetic map of *Aegilops tauschii* chromosome 7D, and physical map of the *Brachypodium distachyon* chromosome 3 (between 39 Mbp and 45 Mbp) using CMap

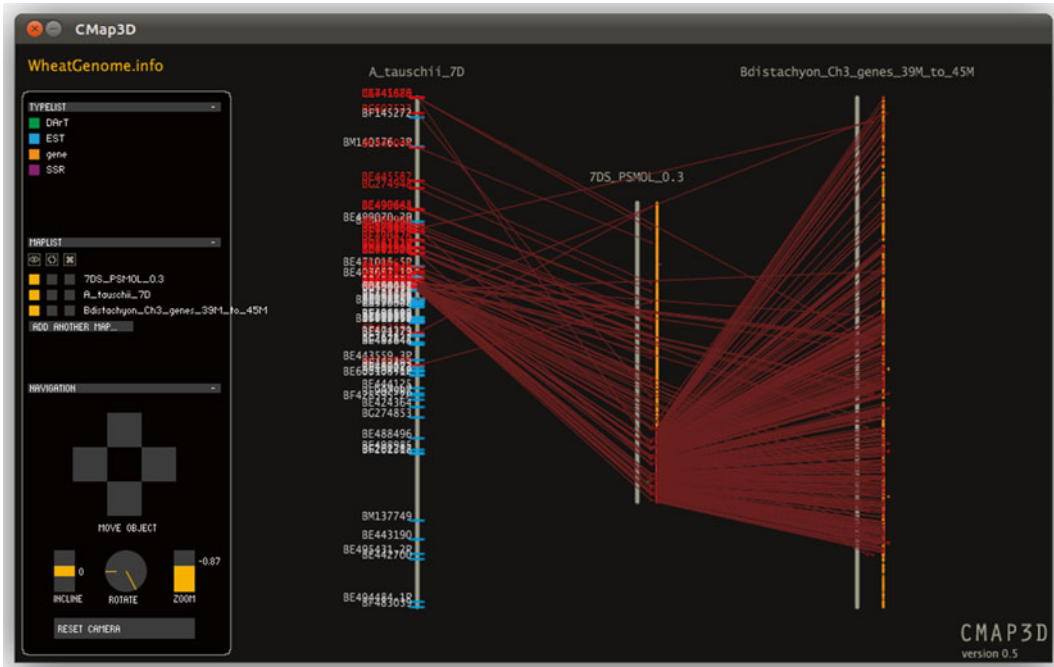


Fig. 6 An interspecies comparison between a physical map of the wheat 7DS syntenic build chromosome, a genetic map of *Aegilops tauschii* chromosome 7D, and physical map of the *Brachypodium distachyon* chromosome 3 (between 39 Mbp and 45 Mbp) using CMap3D [17]

3.7 Wheat autoSNPdb

AutoSNPdb [22, 23] is developed based on autoSNP [24, 25] and SNPserver [26]. AutoSNPdb provides an extensible and user-friendly graphical interface supporting a variety of queries to identify SNP polymorphisms related to specific genes or traits.

The AutoSNPdb portal page provides a drop box for selecting the species. When wheat is selected, AutoSNPdb would process the wheat database. The action list provides links to the main functions of wheat AutoSNPdb.

The first link provides a database search function using keyword(s). The display options are also provided. The second link provides search function for SNPs between selected cultivars (varieties). The user needs to select primary and secondary cultivars and SNPs between these are then displayed. The third link provides a search function for SNPs homologous to reference genomic locations. The user can specify the range and position on the selected chromosome and the resulting SNPs will be displayed. The fourth link provides BLAST searches against the wheat consensus assembly sequences.

The above links display the related consensus assembly sequence results. The search result interface displays list of consensus assembly sequences with annotation associated with the search term. The user can select any of these consensus assembly sequences to view the aligned reads and SNPs in a visual interface (Fig. 7), and related GenBank, gene ontology, and Swiss-Prot annotations.

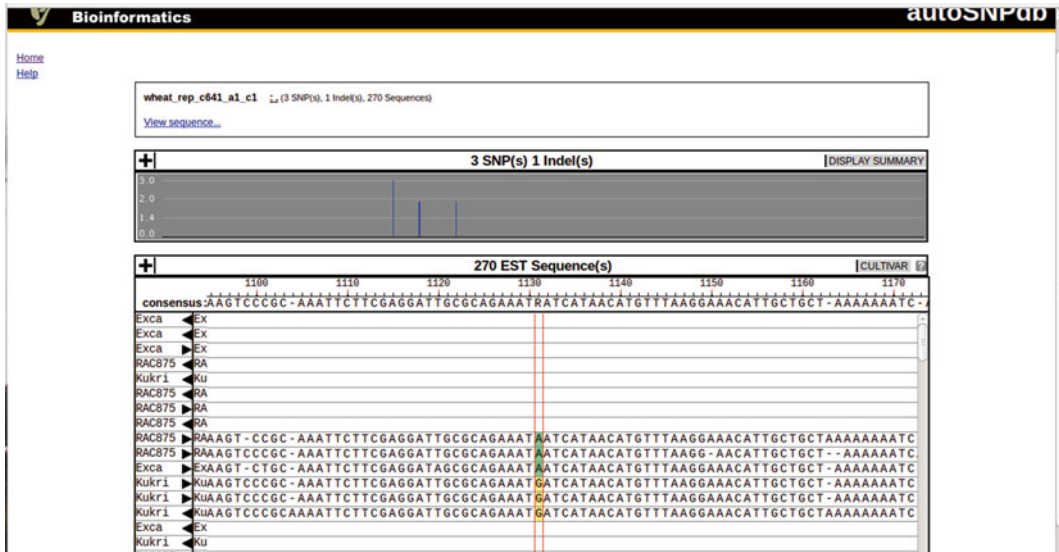


Fig. 7 The wheat autoSNPdb web interface displaying the wheat sequence assembly, and predicted SNPs as vertical bars [17]

References

1. Brautigam A, Gowik U (2010) What can next generation sequencing do for you? Next generation sequencing as a valuable tool in plant research. *Plant Biol (Stuttg)* 12(6):831–841
2. Ansong WJ (2009) Next-generation DNA sequencing techniques. *N Biotechnol* 25(4):195–203
3. Edwards D, Batley J, Snowdon RJ (2013) Accessing complex crop genomes with next-generation sequencing. *Theor Appl Genet* 126(1):1–11
4. Gupta PK, Mir RR, Mohan A, Kumar J (2008) Wheat genomics: present status and future prospects. *Int J Plant Genomics* 2008:896451
5. Berkman PJ, Skarshewski A, Lorenc MT, Lai K, Duran C, Ling EYS, Stiller J, Smits L, Imelfort M, Manoli S, McKenzie M, Kubaláková M, Šimková H, Batley J, Fleury D, Doležel J, Edwards D (2011) Sequencing and assembly of low copy and genic regions of isolated *Triticum aestivum* chromosome arm 7DS. *Plant Biotechnol J* 9(7):768–775
6. Berkman PJ, Skarshewski A, Manoli S, Lorenc MT, Stiller J, Smits L, Lai K, Campbell E, Kubaláková M, Šimková H, Batley J, Doležel J, Hernandez P, Edwards D (2012) Sequencing wheat chromosome arm 7BS delimits the 7BS/4AL translocation and reveals homoeologous gene conservation. *Theor Appl Genet* 124:423–432
7. Berkman PJ, Visendi P, Lee HC, Stiller J, Manoli S, Lorenc MT, Lai K, Batley J, Fleury D, Šimková H, Kubaláková M, Weining S, Doležel J, Edwards D (2013) Dispersion and domestication shaped the genome of bread wheat. *Plant Biotechnol J* 11(5):564–571
8. Somerville C (2006) The billion-ton biofuels vision. *Science* 312(5778):1277
9. Draper J, Mur LA, Jenkins G, Ghosh-Biswas GC, Bablak P, Hasterok R, Routledge AP (2001) *Brachypodium distachyon*. A new model system for functional genomics in grasses. *Plant Physiol* 127(4):1539–1555
10. The International Brachypodium Initiative (2010) Genome sequencing and analysis of the model grass *Brachypodium distachyon*. *Nature* 463(7282):763–768
11. Suzek BE, Huang H, McGarvey P, Mazumder R, Wu CH (2007) UniRef: comprehensive and non-redundant UniProt reference clusters. *Bioinformatics* 23(10):1282–1288
12. Edwards D, Wilcox S, Barrero RA, Fleury D, Cavanagh CR, Forrest KL, Hayden MJ, Moolhuijzen P, Keeble-Gagnère G, Bellgard MI, Lorenc MT, Shang CA, Baumann U, Taylor JM, Morell MK, Langridge P, Appels R, Fitzgerald A (2012) Bread matters: a national initiative to profile the genetic diversity of Australian wheat. *Plant Biotechnol J* 10(6):703–708

13. Lai K, Lorenc MT, Lee H, Berkman PJ, Bayer PE, Muhindira PV, Ruperao P, Fitzgerald TL, Zander M, Chan C-KK, Manoli S, Stiller J, Batley J, Edwards D (2015) Identification and characterisation of more than 4 million inter-varietal SNPs across the group 7 chromosomes of bread wheat. *Plant Biotechnol J* 13(1):97–104
14. Donlin MJ (2009) Using the Generic Genome Browser (GBrowse). *Curr Protoc in Bioinformatics*. Chapter 9 (Unit 9), 9
15. Arnaoudova EG, Bowens PJ, Chui RG, Dinkins RD, Hesse U, Jaromczyk JW, Martin M, Maynard P, Moore N, Schardl CL (2009) Visualizing and sharing results in bioinformatics projects: GBrowse and GenBank exports. *BMC Bioinformatics* 10, A4
16. Donlin M (2007) Using the Generic Genome Browser (GBrowse). *Curr Protoc Bioinformatics*. Chapter 9 (Unit 9), 9
17. Lai K, Berkman PJ, Lorenc MT, Duran C, Smits L, Manoli S, Stiller J, Edwards D (2012) WheatGenome.info: an integrated database and portal for wheat genome information. *Plant Cell Physiol* 53(2), e2
18. Marshall DJ, Hayward A, Eales D, Imelfort M, Stiller J, Berkman PJ, Clark T, McKenzie M, Lai K, Duran C, Batley J, Edwards D (2010) Targeted identification of genomic regions using TAGdb. *Plant Methods* 6:19
19. Zhang Z, Schwartz S, Wagner L, Miller W (2000) A greedy algorithm for aligning DNA sequences. *J Comput Biol* 7(1–2):203–214
20. Youens-Clark K, Faga B, Yap IV, Stein L, Ware D (2009) CMap 1.01: a comparative mapping application for the Internet. *Bioinformatics* 25(22):3040–3042
21. Duran C, Boskovic Z, Imelfort M, Batley J, Hamilton NA, Edwards D (2010) CMap3D: a 3D visualisation tool for comparative genetic maps. *Bioinformatics* 26:273–274
22. Duran C, Appleby N, Clark T, Wood D, Imelfort M, Batley J, Edwards D (2009) AutoSNPdb: an annotated single nucleotide polymorphism database for crop plants. *Nucleic Acids Res* 37:D951–D953
23. Duran C, Appleby N, Vardy M, Imelfort M, Edwards D, Batley J (2009) Single nucleotide polymorphism discovery in barley using autoSNPdb. *Plant Biotechnol J* 7(4):326–333
24. Barker G, Batley J, O'Sullivan H, Edwards KJ, Edwards D (2003) Redundancy based detection of sequence polymorphisms in expressed sequence tag data using autoSNP. *Bioinformatics* 19(3):421–422
25. Batley J, Barker G, O'Sullivan H, Edwards KJ, Edwards D (2003) Mining for single nucleotide polymorphisms and insertions/deletions in maize expressed sequence tag data. *Plant Physiol* 132(1):84–91
26. Savage D, Batley J, Erwin T, Logan E, Love CG, Lim GAC, Mongin E, Barker G, Spangenberg GC, Edwards D (2005) SNPServer: a real-time SNP discovery tool. *Nucleic Acids Res* 33:W493–W495

Chapter 11

User Guidelines for the *Brassica* Database: BRAD

Xiaobo Wang, Feng Cheng, and Xiaowu Wang

Abstract

The genome sequence of *Brassica rapa* was first released in 2011. Since then, further *Brassica* genomes have been sequenced or are undergoing sequencing. It is therefore necessary to develop tools that help users to mine information from genomic data efficiently. This will greatly aid scientific exploration and breeding application, especially for those with low levels of bioinformatic training. Therefore, the *Brassica* database (BRAD) was built to collect, integrate, illustrate, and visualize *Brassica* genomic datasets. BRAD provides useful searching and data mining tools, and facilitates the search of gene annotation datasets, syntenic or non-syntenic orthologs, and flanking regions of functional genomic elements. It also includes genome-analysis tools such as BLAST and GBrowse. One of the important aims of BRAD is to build a bridge between *Brassica* crop genomes with the genome of the model species *Arabidopsis thaliana*, thus transferring the bulk of *A. thaliana* gene study information for use with newly sequenced *Brassica* crops.

Key words BRAD, Brassica database, Genome sequences, Genomic syntenic, Comparative genomics, Brassicaceae, *Brassica* crop

1 Introduction

The *Brassica* database (BRAD) is a Web-based resource focusing on genome-scale genetic and genomic data for Brassicaceae members, especially important *Brassica* crop species [1]. BRAD was initially built using first whole genome sequencing and related datasets of the *Brassica* A genome species *Brassica rapa* [2]. It has since extended to include all available *Brassicaceae* genomes. In general, BRAD provides datasets of complete genome sequences, predicted genes and associated annotations, noncoding RNAs, transposable elements (TE), orthology to *A. thaliana*, genetic markers, and linkage maps of the available *Brassica* genomes. BRAD provides useful search and data mining tools, these include the ability to search across annotation datasets, search for syntenic or non-syntenic orthologs, and search flanking regions of a certain target; it also includes tools such as BLAST, GBrowse, and Synteny visualization. BRAD allows users to search using numerous kinds of

keywords; these include *B. rapa* or *A. thaliana* gene IDs, physical positions, or genetic markers. BRAD is continuously updated with newly available sequenced *Brassica* genomes and related information. There are six major sections of the BRAD website (Fig. 1): Home, Browse, Search, Tools, Download, and Links.

2 User Guidelines for Different Sections of BRAD

2.1 Home: BRAD Homepage

This section provides a brief introduction of BRAD and the direction of its future development (Fig. 2). The “News section” in the right sidebar is designed to inform users of the latest BRAD updates. The information page linked from “Contact” allows the user to contact the support team to find help with any questions they have related to using BRAD.

2.2 Browse: Genomic Markers and Phenotypes of *B. rapa*

BRAD provides genetic markers of *B. rapa* [3] for map construction in different *B. rapa* populations; it also includes gene families, five groups of genes related to different metabolic pathways, transcription factors, and morphological diversity of *B. rapa* phenotypes. A brief introduction to the Wang lab is also given (Fig. 3). The “Browse overview” webpage provides more detailed information of this section.

2.2.1 Markers and Maps

For each marker present in the “Browse” genetic markers and maps section, the genetic and physical position, primer information, and parental populations are provided. Accession to data follows the order: chromosome selection → population specification → detailed marker information → click on marker ID for primer information. An example of clicking Chr (chromosome) A01 on the picture or on the menu in the left sidebar is shown in Fig. 4, and the output result shown in Fig. 5. Information on population, chromosome, and marker number is provided in the output table. After clicking on marker number “34” of *B. rapa* population RCZ16_DH a list is given of these markers (Fig. 6). More detailed information for each marker, such as primer sequence, length, and position is obtained by clicking on the marker ID, e.g., KBRH139B23-1 (Fig. 7).

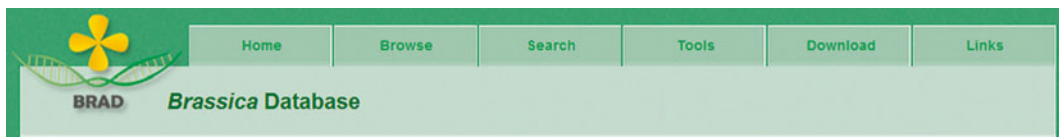


Fig. 1 BRAD navigation. There are main six main sections navigable from the BRAD menu bar: Home, Browse, Search, Tools, and the resources: Download, and Links



Fig. 2 Homepage. This section provides a short introduction to BRAD, and features daily news updates on the right sidebar

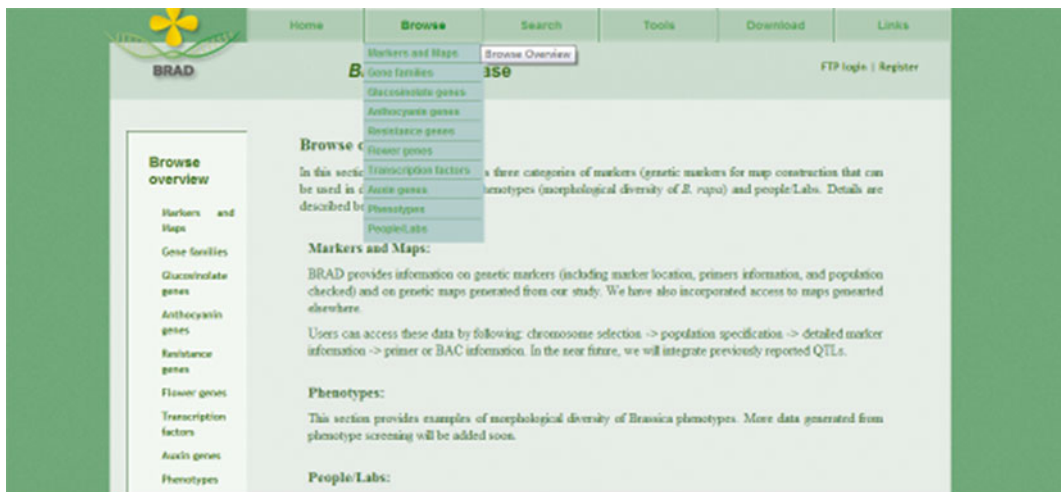


Fig. 3 Browse. This section of BRAD gives access to datasets of gene families, genetic markers, and maps of *B. rapa*

2.2.2 Genes of Important Pathways or Function Families

Glucosinolate genes [4], anthocyanin genes [5], resistance genes, flowering-related genes [6], auxin-related genes, and the main gene families and transcription factors can be determined from the *B. rapa* genome. Users are required to follow the same three steps outlined in Subheading 2.2.1. Figure 8 shows 30 auxin-related genes belonging to the ARF gene subfamily. Clicking the gene ID, e.g., “*Bra013748*” will pop out a search navigation window providing links to the following gene information: Annotations, Genome browse, Syntenic paralogs, Non-syntenic At-Br orth,

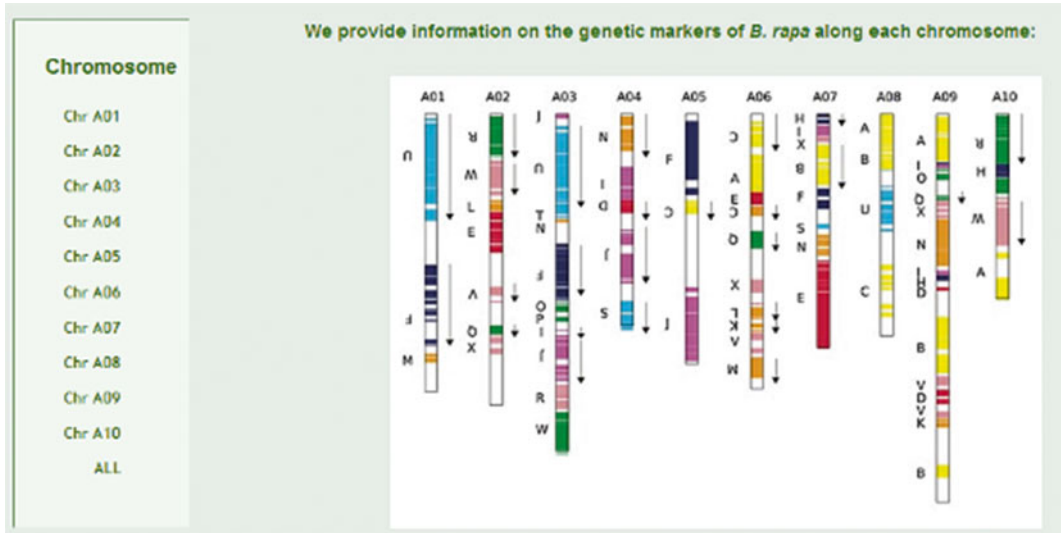


Fig. 4 Distribution of genomic blocks on ten chromosomes of *B. rapa*. Clicking each bar retrieves marker information for the corresponding chromosome

The genetic marker in *B. rapa* chromosome A01

Population	Chromosome	Marker Number
RCZ16_DH	A01	<u>34</u>
JWF3P	A01	<u>33</u>
VCS_DH	A01	<u>21</u>
Total	A01	<u>88</u>

Fig. 5 Output table of *B. rapa* population, chromosome, and marker number information for chromosome A01

Gene sequence, Flank regions, and accessions to the CoGe (comparative genomics) website (<https://www.genomeevolution.org/CoGe/>). Detailed information of these functions is illustrated in a later section of this report.

2.3 Search: Annotations, Syntenic Genes, and Genomic Regions

This section is designed to annotate predicted genes in *Brassica* and to give users a comprehensive understanding of any genes of interest. There are five main parts in the “Search” section: annotations, syntenic gene pairs, non-syntenic At-Br orthologues, element of flanking regions, and gene and protein sequences.

2.3.1 Annotations

Seven kinds of annotation datasets are provided for the predicted genes of *B. rapa*: Swissprot annotation, Trembl annotation [7], KEGG annotation [8], InterPro domain annotation [9], Gene Ontology of *Brassica* [10], and orthologous genes (best hit of

The genetic marker in *B. rapa* chromosome A01

Chr	Marker	Genetic position	Physical position	Strand	Population
A01	KBRH139B23-1	0	2507626 - 2507722	+	RCZ16_DH
A01	Ra2G09	2.377	3282650 - 3282886	+	RCZ16_DH
A01	BrID10277	3.907	4324694 - 4324786	+	RCZ16_DH
A01	BrID10279	7.62	4177408 - 4177488	+	RCZ16_DH
A01	BRMS056(R1)	8.653	4934143 - 4934358	+	RCZ16_DH
A01	BrID10297	9.182	4634051 - 4634146	+	RCZ16_DH
A01	BrID10299	9.358	4723241 - 4723328	+	RCZ16_DH
A01	BrID10985	12.062	5717292 - 5717413	+	RCZ16_DH
A01	BrID10301	12.693	5580536 - 5580622	+	RCZ16_DH
A01	BrID10303	13.491	6163928 - 6164027	+	RCZ16_DH
A01	BrID10307	19.926	8554140 - 8554239	+	RCZ16_DH
A01	BrID101037	20.776	8004498 - 8004631	+	RCZ16_DH
A01	BrID10305	40.264	7527860 - 7527947	+	RCZ16_DH
A01	BC46(R1)	16.005	0 - 0	.	RCZ16_DH
A01	OH12F11(R1)	18.45	0 - 0	.	RCZ16_DH
A01	BrID101119	19.926	9895949 - 9896088	-	RCZ16_DH
A01	BrID10823	26.894	12086918 - 12087016	-	RCZ16_DH
A01	BrID90367	45.841	12403388 - 12403565	-	RCZ16_DH
A01	BrID10771	27.099	14044461 - 14044552	-	RCZ16_DH
A01	BrID101187	27.179	16062865 - 16062955	-	RCZ16_DH
A01	BrID101211	27.274	12614564 - 12614694	-	RCZ16_DH
A01	BrID10961	27.688	13394163 - 13394302	+	RCZ16_DH

Fig. 6 Marker names, genetic position, physical position, and strand information for the *B. rapa* population RCZ16_ZH

BLASTX) between *B. rapa* and *A. thaliana* (Fig. 9). These datasets can be used to annotate gene models from different aspects, such as nucleotide sequence, proteins, and domains. In the annotation search section, users can find genes with functions of interest through submitting a keyword, such as flower, growth, or gene ID; this will collect genes with related functions from the seven aforementioned annotation datasets. As an example: type in the *B. rapa* gene ID “Bra019255” and then click on the GO button. Further clicking on the result of an annotation dataset will provide annotation information of genes corresponding to the searched keyword. Figure 10 gives an example of an output table following such an event; it lists six Gene Ontology Annotation search hits relating to “Bra019255.” Clicking on the gene ID in the output table will pop out a search navigation menu, while clicking on a Gene Ontology ID will take the user to the AmiGO website [11] for more detailed annotation information.

KBRH139B23-1 Details

Name: KBRH139B23-1
Type: geneticMarker
Source: RCZ16_DH
Position: A01:2507626..2507722 (+ strand)
Length: 97
Score: 0
PrimerF: ATCTCATGGTTGGTTCACCG
PrimerR: ATTTCCAAAACACACACGCA
load_id: KBRH139B23-1
primary_id: 771067
gbrowse_dbid: BrapaV1.2:database

```

>KBRH139B23-1 class=Sequence position=A01:2507626..2507722 (+ strand)
ATCTCATGGT TGGTTCACCG TGAGATCCAT GGAGAGAACG AACCAAGGACG TGTTGGAGGA GGAGGAGGAG GAGGAGTGC
GTGTGCGTTT TGGAAAT
    
```

Fig. 7 Detail information for the KBRH139B23-1 marker of the *B. rapa* population RCZ16_DH

Auxin genes of sub family ARF in *B. rapa*
Gene ID of A. thaliana followed by a bracket 's' means syntenic ortholog

Index	Gene (<i>B. rapa</i>)	At ortholog
1	Bra013748	AT4G23980 (s)
2	Bra035427	AT1G59750 (s)
3	Bra014419	AT3G61830 (s)
4	Bra011955	AT2G28350 (s)
5	Bra034234	-
6	Bra003665	AT1G77850 (s)
7	Bra002327	AT5G20730 (s)
8	Bra010048	AT5G62000 (s)
9	Bra029293	AT5G62000 (s)
10	Bra017362	-
11	Bra018124	-
12	Bra015374	-
13	Bra024109	AT4G30080 (s)
14	Bra016492	AT1G19850 (s)
15	Bra028110	AT5G37020 (s)
16	Bra005465	AT2G33860 (s)
17	Bra025775	AT1G19850 (s)
18	Bra011162	AT4G30080 (s)
19	Bra015651	AT1G77850 (s)
20	Bra020125	AT5G20730 (s)
21	Bra007632	AT3G61830 (s)

Fig. 8 Genes belonging to the auxin sub family ARF in *B. rapa*, and their corresponding paralogs in *A. thaliana*

Search across annotation datasets

Bra019255 [Help](#)

examples: AUXIN, ATP, AT4G23980, Bra019255

Display records/page

Gene Ontology annotation [6 records](#)

InterPro domain annotation [4 records](#)

KEGG annotation [1 records](#)

Swissprot annotation [1 records](#)

Trembl annotation [1 records](#)

Orthologous genes to *A. thaliana* [1 records](#)

BLASTX (best hit) to *A. thaliana* [1 records](#)

Fig. 9 The gene annotation search interface. BRAD includes seven different gene annotation datasets. Search results for the *B. rapa* gene *Bra019255* are shown in the above example

Results

[First page](#) | [Up page](#) | [Next page](#) | [Last page](#)

The information of Gene Ontology annotation by keyword: Bra019255

Gene	Gene Ontology
Bra019255	GO:0046983 ; protein dimerization activity; Molecular Function
Bra019255	GO:0045449 ; regulation of transcription; Biological Process
Bra019255	GO:0009725 ; response to hormone stimulus; Biological Process
Bra019255	GO:0006355 ; regulation of transcription, DNA-dependent; Biological Process
Bra019255	GO:0005634 ; nucleus; Cellular Component
Bra019255	GO:0003677 ; DNA binding; Molecular Function

[First page](#) | [Up page](#) | [Next page](#) | [Last page](#)

Fig. 10 Gene ontology (GO) of the *B. rapa* gene *Bra019255*. Clicking on a GO item directs the user to the AmiGO database

2.3.2 Syntenic Gene

The syntenic relationship of genes between *A. thaliana* and three *B. rapa* subgenomes can be searched using gene IDs from either species. The Syntenic gene web search section (Subheading 2.3.2) gives two options for number of flanking genes in the pull-down menu, allowing 10 or 20 genes flanking both sides of the searched gene to be displayed. For example, typing in the gene ID “AT4G23980” and setting the flanking gene number to 10 will

output the table shown in Fig. 11. The heading of the output table includes abbreviations as “LF,” which denotes the least fractioned subgenome in *B. rapa*, while “MF1” means the medium fractionated subgenome and “MF2” represents the most fractionated subgenome [2, 12, 13]. The targeted gene is shaded dark green in the center of the table, and each *A. thaliana* gene corresponds to 1, 2, or 3 genes in the three subgenomes of *B. rapa*; “-” indicates that the gene was fractionated. Hovering the mouse over a gene ID displays the function annotations of *A. thaliana* genes, and provides detailed supporting information on the synteny relationships of *B. rapa* genes to *A. thaliana*. Clicking on gene IDs listed in the output table triggers a small pop out window for search navigation, while clicking on the tandem symbol “-Tandem” shows gene

Search syntenic genes between *A. thaliana* and subgenomes of *B. rapa*.

AT4G23980 Flanking genes. [Help](#)

examples: AT4G23980, Bra019255, Bra019257

Results <-Backword Forward->

Searching of AT4G23980

<i>A. thaliana</i>	LF of <i>B. rapa</i>	MF1 of <i>B. rapa</i>	MF2 of <i>B. rapa</i>
AT4G23895 U	-	Bra019258	Bra010547
AT4G23900 U	Bra013742	-	-
AT4G23910 U	Bra013743	-	-
AT4G23920 U	Bra013744	Bra019257	-
AT4G23930 U	Bra013745	Bra019256	-
-	Bra013746	-	-
AT4G23940 U	Bra013747	-	-
AT4G23950 U	-	-	-
AT4G23960 U	-	-	-
AT4G23970 U	-	-	-
AT4G23980 U	Bra013748	Bra019255	-
AT4G23990 U - Tandem	Bra013749 - Tandem	Bra019254	-
-		Bra019253	-
AT4G24015 U	Bra013751	Bra019252	-
-	Bra013752	-	-
-	Bra013753	-	-
AT4G24020 U	Bra013754	Bra019251	-
AT4G24030 U	-	-	-
AT4G24026 U	Bra013755	Bra019250	-
AT4G24040 U	Bra013756	Bra019249	-
AT4G24050 U	Bra013757	Bra019248	-

ARF9; ARF9 (AUXIN RESPONSE FACTOR 9); transcription factor

Fig. 11 Searching syntenic genes of *AT4G23980* between *A. thaliana* and *B. rapa* subgenomes. Mousing over the gene ID shown in the output table displays the gene annotation information for the *A. thaliana* gene and alignment information of the *B. rapa* gene

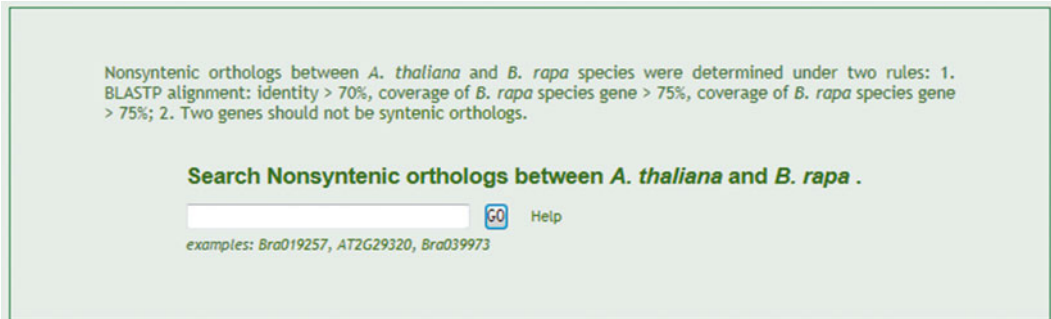
members of the tandem array at the bottom of the searching outputs.

2.3.3 Non-syntenic At-Br Orth

The datasets of non-syntenic gene pairs between *A. thaliana* and *B. rapa* provided here were determined under two rules. First, the parameters of BLASTP alignment: identity > 70 %, coverage of *A. thaliana* gene > 75 %, and coverage of *B. rapa* gene > 75 %. Second, two genes of an orthologous pair should not be a syntenic gene pair. A total of 17,159 non-syntenic orthologous pairs were determined. Figure 12 shows the search page for non-syntenic At-Br Orth. For example, a search for the “Bra019257” gene will obtain the results displayed in Fig. 13. Detailed alignment information is shown in the output table, allowing users to consider homology level; a search navigation window is linked from the gene IDs output in the table.

2.3.4 Searching the Flanking Region

This module was developed to help users locate functional elements distributed in the flanking region of a target (Fig. 14). The target can be a physical position, a gene ID, or a genetic marker. All genomic elements, such as genes, transposons, and RNAs



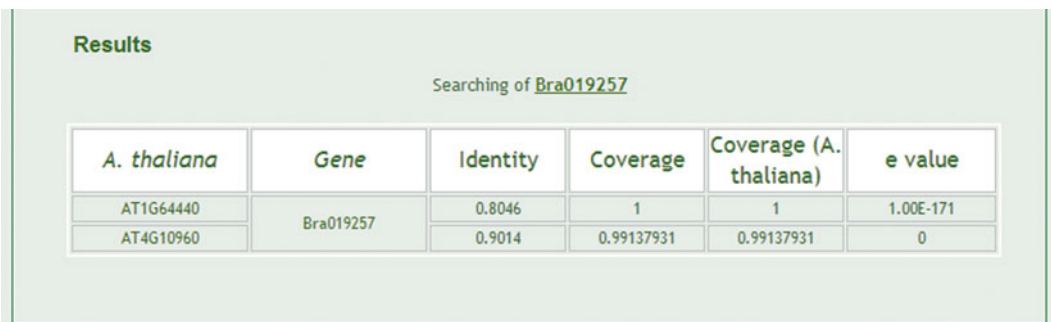
Nonsyntenic orthologs between *A. thaliana* and *B. rapa* species were determined under two rules: 1. BLASTP alignment: identity > 70%, coverage of *B. rapa* species gene > 75%, coverage of *A. thaliana* species gene > 75%; 2. Two genes should not be syntenic orthologs.

Search Nonsyntenic orthologs between *A. thaliana* and *B. rapa* .

[GO](#) [Help](#)

examples: Bra019257, AT2G29320, Bra039973

Fig. 12 The non-syntenic At-Br orthologs search window



Results

Searching of [Bra019257](#)

<i>A. thaliana</i>	Gene	Identity	Coverage	Coverage (<i>A. thaliana</i>)	e value
AT1G64440	Bra019257	0.8046	1	1	1.00E-171
AT4G10960		0.9014	0.99137931	0.99137931	0

Fig. 13 Search result display of non-syntenic At-Br orthologs

The screenshot shows a web interface for searching flanking regions. It is divided into two main sections: 'Search input' and 'Target item'. In the 'Search input' section, there is a text input field, a 'flank' dropdown menu currently set to '2', a 'kb' unit, and a 'GO' button. Below the input fields, there are example search terms: 'examples: A01:801000..825000, Bra000009, BrID10419'. The 'Target item' section contains four checkboxes: 'Gene' (which is checked), 'RNA (miRNA, tRNA, rRNA, snRNA)', 'Transposon', and 'Genetic Marker'.

Fig. 14 The flanking region search window

(miRNA, tRNA, rRNA, and snRNA) located in the flanking regions of the target are collected and displayed in an output table. For example, searching with the *B. rapa* marker “BrID10419” will output the genomic elements results shown in Fig. 15. The tabular data contains the start position, stop position, and chromosome strand of each element, and provides population information of the genetic markers. The first line of the search result provides a link to GBrowse, which helps users to visualize the target region under the background of the corresponding chromosome. This is a useful tool for studies that involve fine mapping of target genes in QTL analysis. Once QTLs have been obtained, markers can be aligned to genome sequences to obtain their physical positions. Elements in the flanking regions of these markers can thus be checked to locate candidate genes or miRNAs that may be causal factors of the QTLs.

2.3.5 Gene Sequence

This interface (Fig. 16) is built to help users easily obtain nucleotide or protein sequences by entering their gene ID (e.g., “*Bra019255*”; Fig. 17). The first line of the output result provides a link to the GBrowse interface, and below this is a table listing gene positions. Clicking on “Gene sequence” or “Protein Sequence” links will bring up a new webpage with the complete corresponding sequence of the searched gene (Fig. 18).

2.4 Tools: Blast, Genome Browse

BLAST and GBrowse [14] are embedded in BRAD to help users align and visualize the genomic datasets.

Results

Searching of BrID10419 (A02:6913088..6923181)

Item	Name	Chromosome	Start	Stop	Strand
Gene	Bra022699	A02	6921347	6921538	-
Gene	Bra022700	A02	6917370	6919232	-

Item	Chromosome	Start	Stop	Strand
miRNA		None		
tRNA		None		
rRNA		None		
snRNA		None		

Item	Chromosome	Start	Stop	Strand
Transposon	A02	6914730	6914958	+
Transposon	A02	6915221	6915339	-
Transposon	A02	6915354	6915703	-
Transposon	A02	6918263	6918310	+
Transposon	A02	6918275	6918616	+
Transposon	A02	6918617	6918814	-
Transposon	A02	6919189	6919337	-
Transposon	A02	6919290	6919344	-
Transposon	A02	6920693	6920776	+
Transposon	A02	6920871	6920979	+
Transposon	A02	6921203	6921614	-
Transposon	A02	6922274	6922446	-
Transposon	A02	6922312	6922455	-
Transposon	A02	6922656	6922900	-
Transposon	A02	6922943	6923136	-

Item	Name	Chromosome	Start	Stop	Strand	Population
Marker	BrID10419	A02	6918088	6918181	+	RCZ16_DH

Fig. 15 Genomic elements in the flanking region of the *BrID10419* marker. Target items including genes, RNAs, transposons, and genetic markers are shown in the output table

Search gene sequence and physical position of *Brassicaceae* species by gene ID.
 Genomic sequence can be downloaded in Gbrowse (Option in Reports & Analysis) flexibly and freely.


 eg: Bra019255

Fig. 16 Gene sequence search window

2.4.1 BLAST

The standard `wwwblast` (<http://www.ncbi.nlm.nih.gov/staff/tao/URLAPI/wwwblast/>) module was adopted by BRAD to provide a BLAST service for users. Numerous databases were collected for BLAST, including genome, gene, and protein sequences of *B. rapa*; EST sequences of *B. rapa*, *Brassicaceae*, and *Brassicaceae* (*Cruciferae*) (Fig. 19). Using the “*Bra019255*” gene sequence as an example, when the database of *B. rapa* genes is selected with

Results

Searching of [Bra019255](#)

Name	Chromosome	Start	Stop	Strand	Gene	Protein
Bra019255	A03	25446636	25449715	+	Gene Sequence	Protein Sequence

Fig. 17 Gene search result for the *Bra019255* gene

```
>Bra019255
MYGELWKLKAGPVVDVPQAAERVFYFPQGHMEQLEASTQQDLNAVKPTKP
LFDLPPKILCRVMDVRLQAEKDTDEVYAQIMLMPEGTVDEPMSPDFSPP
E SQRPKVHSFSKVLTA SDTSTHGGFVLRKHATECLPPLDMTQQTPTQE
LV AEDVHGYQWKFKHIFRGQPRRHLLTTGWSTFVTAKRLVAGDTFVFLR
GEN GELRVGVRANRQQTNPSSVISHSMHLGVLATACHATQTRSMFT
VYYK PRTSQFIIISLNKYLEAMSNKFSVGIRFKMRFEGEDSPERRLSGTV
GGGKD CSTHWKDSNWRCLEVHWDEPASISRDKVSPWEIEPFVTSENVP
HSVMPK NKRPRHYSEVSALGKTASNLSWSSALTSHEFAQSCITSQRNSP
QQCYRDA TEDAKNSDWSASPYSATLNNQMVFPVEQKKPETTASYRLFGI
DLLSSSIP ATEEKTAPTLPINITKPTPDSNSDPKSEVSKLSEEKKQEP
A QASSKEVQS KEISSTRSRTKVQMVGVPVGRAVDLTVLNGYSELIDDLEK
LFDIEGELKS RNQWEIVFTDDEGDMMLVGGDPWPEFCNMVKRIFIWSKE
EVKMTPGNQL RLLTEVDTTLTITISKTENHSN
```

Fig. 18 Retrieved protein sequence for the *Bra019255* gene

default parameters to run BLAST, there are 83 hits of the query sequence (Fig. 20).

2.4.2 Genome Browse (GBrowse)

GBrowse visualizes genomic elements (including genes, noncoding RNAs, TEs, and genetic markers) of the *B. rapa* genome on one frame. Links from the genes visualized in GBrowse direct users to other BRAD applications. Tracks shown in the detailed section of GBrowse are gene models derived from *B. rapa* genome version 1.01, and genomic elements of mRNA, CDS, genetic marker, transposon, miRNA, tRNA, snRNA, rRNA, and SSR (Fig. 21). Clicking on a gene icon in GBrowse returns links to its annotations, the best BLASTX match to *A. thaliana*, and the function and Gene Ontology (GO) of the matched *Arabidopsis* gene (Fig. 21). Additionally, the user can choose the option “Download Sequence File” in the menu of the configure panel (Fig. 22) to

BLAST

BLAST program:

Data DB:

Input:

Or load it from disk

Set subsequence: From To

Other Options

Filter: Low complexity Mask for lookup table only

Expect: Matrix: Perform ungapped alignment

Frame shift penalty for blastx:

Alignment view:

Other advanced options:

• Descriptions: Graphical Overview

• Alignments: Color schema:

Fig. 19 Webpage for the BLAST service provided in BRAD

retrieve the sequence of any genomic region required; an example is shown in Fig. 23.

2.5 Download: Brassica Plant Datasets

In addition to being able to download genomic data from GBrowse, this section also provides links for bulk data downloads. Data options include genome and gene sequences, gene annotations, and other predicted genomic elements (Fig. 24). Other *Brassica* plant datasets will be provided in BRAD as soon as they are made available. These datasets will be updated when necessary changes are made to genome references.

2.6 Links: Other Brassicaceae Resources

BRAD makes numerous community resources available, either as data or through external website links. These resources include

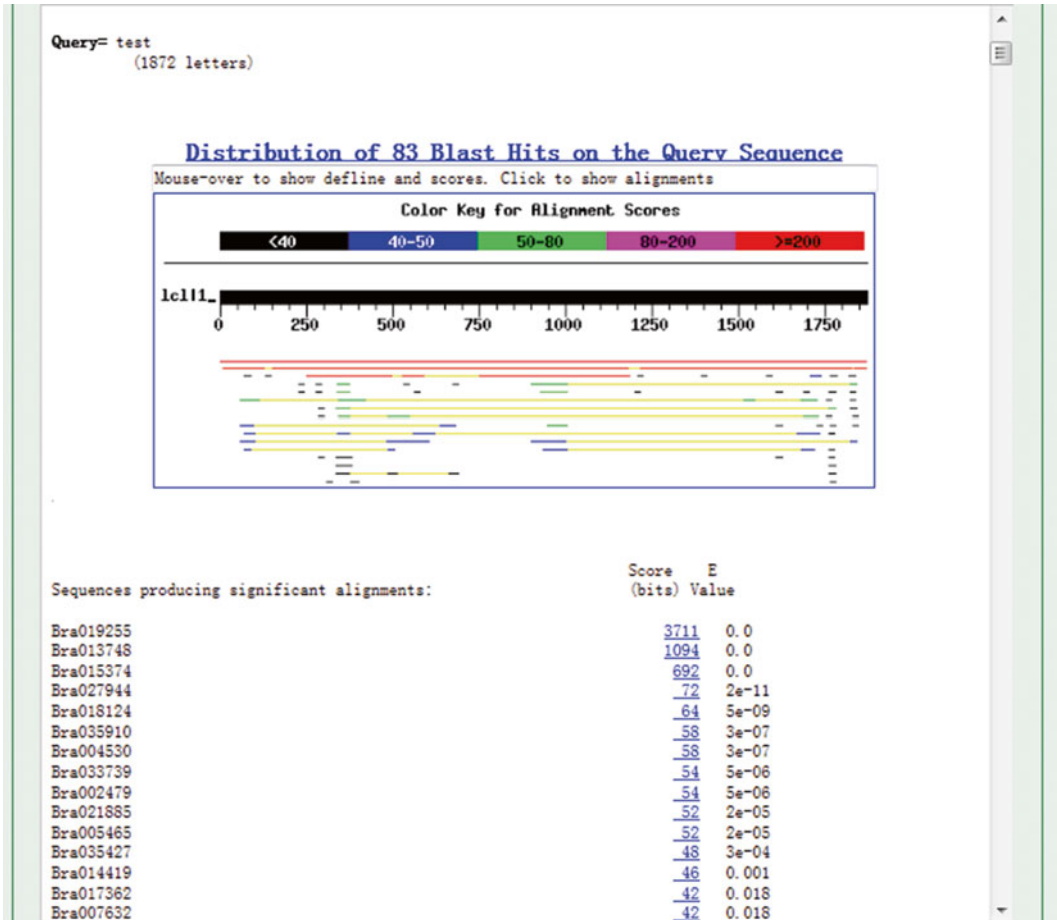


Fig. 20 BLAST alignment of the *Bra019255* gene against the *B. rapa* gene database

other databases or web resources from laboratories focusing on Brassicaceae studies, and information on meetings of potential interest to *Brassica* researchers and breeders.

3 Discussion and Future Development

BRAD is a database built for studies focusing on the genetics and genomics of *Brassica* plants. It has specific functions, and advantages over other databases of *Brassica* plants, specifically its annotation and deep mining of the *B. rapa* genome, and its use of the gene information from *A. thaliana*. It is aimed at helping scientists and breeders to use the genomic and genetic datasets of *Brassica* plants with utmost ease and efficiency. BRAD will continuously improve its search and visualization tools, and will continue to integrate even more related datasets in the future.

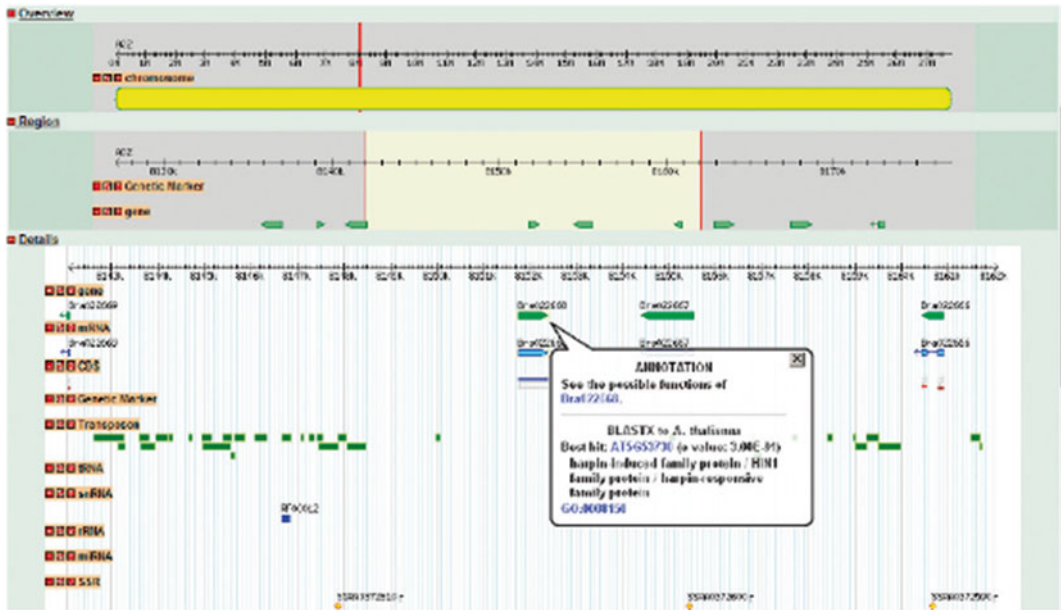


Fig. 21 Genome visualization using Gbrowse. Tracks shown in the detailed section are gene models, mRNA, CDS, genetic marker, TEprotein, transposon, miRNA, tRNA, snRNA, rRNA, and SSRs of *B. rapa* genome version 1.0. Clicking on the icon of the gene model provides a contextual menu with links to annotations and its best-hit gene (BLASTX) in *A. thaliana*. Clicking on other elements will lead users to detailed annotation and sequence information

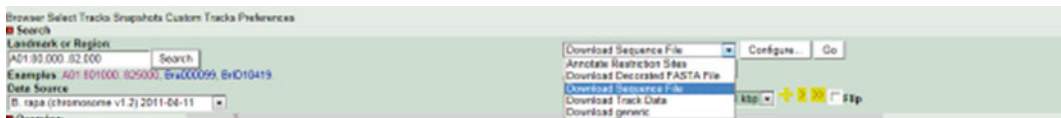


Fig. 22 The Genome Browse panel. Gene ID, gene marker, and genome region are accepted as keywords for searches

A01:80000..82000

```

>A01:80000..82000
TTGTGGTTGACTGTAGCTGATTCCCTCAATCTCAATGCATTGAACTTCTCTGCAATATTCT
TCTGATGGATCTTCTGAACGCGTAGGCAGCTCTTTCATCATCATCATCTCAGAATGAGAC
CCAACATAAGCTCTTGTATTGATGTCTGTGACACTGAACCATCTCCCACTTGTCAATG
TGCTGCGGAGTTCACACTTCAACAGAGTAGAAGCAAAAATAATGTGAATATATTTTGTG
CATAAAGTGAAATTGTGGGAAAAATAAATACCTTTGAGGCCTCATCGAGTTC AACCATTC
TCATGATATCTTCAAGCCGGGATTGAGCAAGATCTCTGTCTTCTCAACTTGCCTCTCT
CCTTTTCCATCTGCATTCAACATTACACCTTAGGTTAGCAATTAACCCAGCCTGAAAAGGA
GATCTATCATAAGCACAAACCTTTTGTATCTGAAGATTCTTCTTCTCACCCTCATCGCA
CAATCACATTTTCGAGGCAGGCGTGGCAGGGGTTCTCAGCTCGGTCTCGAGCCTCGCAAGC
TCACGCTGTAGTTGCTGCAGCAGGGCCTTGTCTGACATAACGACGTTGATCCGGGCTTT
GTGGTAACTTCTTTGCACAGCAAGCAAAACAAGAGAGTGTTCTCGTTAGCTCGACGTGA
CTCTCGTGGGCTCAGAGTGCAGATGATGGCAGTTCCTTGAATCCCACCCAAGCATGGC
TGTAAGATTCGTGTGAGCTTGGAGTCTCTAAAGTTGATGTGCCTTGCCCTTCCCTTACTA
TATAGATCGATCACCAGTGCTTAGACAATTAAGTTAATCTACTAAGGATGTATTGTTT
ACACACCTAAGTTTACGGATCACAGTTCACAGTTCCAAGAGTAAGCAAACTTCGGTTGATGGCAG
CCTTCTTGCAGCCTCGCACCAGCTGACATCGCTGTGATGCAGCTCGCTTCCGCCAGA
TCTATGAAATTCGCACGTTAAAGGAAGAATAATGAACAGAAAAACATTAACACCCCAAGG
ATCCAACATAAAAATAAAGGGTTGTTGATGAGAGAGCATACCACTCGCCATGAGGGTGG
TGGAGTTTTCTTTGTCTAAGAACTCACGAGCAGAGCTTCAACCCGTCTATAAAAAATACAT
TCAAAGCCAAAGCTAAGTAAAAGAGAATCATGGTATCTTAAATACAGTGATAGATAAAG
AGTTTTAACCACTAATAATCTGATGAGATCTGGAACCTCTCTCATTCACTGAGGTTTC
ACCAATCTTACGTTGTGCTGCCAAAAATTAATGAGACATTTAAGTTACAACATTAATCTG
AATAACAACTCTGATTATATTATATCATATACATATATATATATATATATATATATGCT
ACTTCATAGAAATGCAAAGTGATTTTTGACCTTCACAACAGATAGAAGGTCCTTGAGAT
GGTTCCAATCTCGCAGAGTTTCTCTGTGGCTTTTTCAACCACTGTCCCTTCTAGTAAAG
AACCAAGAAAGAACAAAGAGATTGAGACAAAATTTACAGAGCGCTCAGTAAAGAAATTTG
ACATGAAAAATTTTACCTCAGGATCGTCTCGTAGCCTAAGGGATGTACCATCGGAGCTGA
GCAATCTCGGATGGCCTCATTGTAGATCTCTATAGCTGAAAATTTAACGGAAAATGCTC
TTTCTCATGCTACAGACAGCATCCATATTCAATTAGTATTGGGTGCAGCAGTCAGAACCC
AAAAAAAAAAAAAAAAATGATATATTGGAGAAGCCTGATTGCCTGAAAAATGTAGTTAAA
AATGTACGCCACAGCAAACCTCAGTGATAGCAGACATGGTGTAAAGTCTTCCGCTACTCGT
CTGGCCGTACGCCAAAAATACTACCTGTCAAAAACGTTACAGTAAAATGTGAGCATATCGG
TTTCATTATCACCAGATAAGCATATTTGAAGGAAGATTACGTACAATTGATTCCTTTG
ACAACAGAGAGAGCGATGTCC

```

Fig. 23 Retrieved DNA sequence between 80,000 bp and 82,000 bp on chromosome A01 in *B. rapa*



Fig. 24 Bulk data download webpage of BRAD

References

1. Cheng F, Liu S, Wu J, Fang L, Sun S, Liu B, Li P, Hua W, Wang X (2011) BRAD, the genetics and genomics database for *Brassica* plants. *BMC Plant Biol* 11(1):136
2. Wang X, Wang H, Wang J, Sun R, Wu J, Liu S, Bai Y, Mun J-H, Bancroft I, Cheng F (2011) The genome of the mesopolyploid crop species *Brassica rapa*. *Nat Genet* 43(10):1035–1039
3. Wang Y, Sun S, Liu B, Wang H, Deng J, Liao Y, Wang Q, Cheng F, Wang X, Wu J (2011) A sequence-based genetic linkage map as a reference for *Brassica rapa* pseudochromosome assembly. *BMC Genomics* 12(1):239
4. Wang H, Wu J, Sun S, Liu B, Cheng F, Sun R, Wang X (2011) Glucosinolate biosynthetic genes in *Brassica rapa*. *Gene* 487(2):135–142
5. Guo N, Cheng F, Wu J, Liu B, Zheng S, Liang J, Wang X (2014) Anthocyanin biosynthetic genes in *Brassica rapa*. *BMC Genomics* 15(1):426
6. Srikanth A, Schmid M (2011) Regulation of flowering time: all roads lead to Rome. *Cell Mol Life Sci* 68(12):2013–2037
7. Boeckmann B, Bairoch A, Apweiler R, Blatter M-C, Estreicher A, Gasteiger E, Martin MJ, Michoud K, O'Donovan C, Phan I (2003) The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Res* 31(1):365–370
8. Kanehisa M, Goto S (2000) KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res* 28(1):27–30
9. Hunter S, Apweiler R, Attwood TK, Bairoch A, Bateman A, Binns D, Bork P, Das U, Daugherty L, Duquenne L (2009) InterPro: the integrative protein signature database. *Nucleic Acids Res* 37(Suppl 1):D211–D215
10. Consortium GO (2004) The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Res* 32(Suppl 1):D258–D261
11. Carbon S, Ireland A, Mungall CJ, Shu S, Marshall B, Lewis S (2009) AmiGO: online access to ontology and annotation data. *Bioinformatics* 25(2):288–289
12. Cheng F, Wu J, Fang L, Sun S, Liu B, Lin K, Bonnema G, Wang X (2012) Biased gene fractionation and dominant gene expression among the subgenomes of *Brassica rapa*. *PLoS One* 7(5):e36442
13. Tang H, Woodhouse MR, Cheng F, Schnable JC, Pedersen BS, Conant G, Wang X, Freeling M, Pires JC (2012) Altered patterns of fractionation and exon deletions in *Brassica rapa* support a two-step model of paleohexaploidy. *Genetics* 190(4):1563–1574
14. Stein LD, Mungall C, Shu S, Caudy M, Mangone M, Day A, Nickerson E, Stajich JE, Harris TW, Arva A (2002) The generic genome browser: a building block for a model organism system database. *Genome Res* 12(10):1599–1610

Chapter 12

TAG Sequence Identification of Genomic Regions Using TAGdb

Pradeep Ruperao

Abstract

Second-generation sequencing (SGS) technology has enabled the sequencing of genomes and identification of genes. However, large complex plant genomes remain particularly difficult for de novo assembly. Access to the vast quantity of raw sequence data may facilitate discoveries; however the volume of this data makes access difficult. This chapter discusses the Web-based tool TAGdb that enables researchers to identify paired read second-generation DNA sequence data that share identity with a submitted query sequence. The identified reads can be used for PCR amplification of genomic regions to identify genes and promoters without the need for genome assembly.

Key words Second-generation sequencing, Orthologous sequences, Illumina, Promoter discovery

1 Introduction

New sequencing technologies outperform Sanger-based sequencing in throughput and cost [1]. Second-generation sequencing (SGS) platforms such as those from *Roche (454)*, *Illumina*, *Pacific Biosciences*, *Ion Torrent*, and third-generation sequencing (TGS) such as *Nanopore* Sequencing technologies produce large quantities of data. While this rapid advancement of DNA sequencing technologies revolutionizes research methods, it also produces new computational challenges.

Illumina SGS data takes the form of huge numbers of short sequence pair reads with known orientation and estimated insert size. Highly redundant sequence (high coverage) data is needed for assembly to represent a genome. Additionally, the storage and interrogation of this data is becoming an increasing challenge [2, 3].

The ability to search the huge quantity of this SGS data is made feasible by the development of the custom database TAGdb. This chapter discusses TAGdb for the identification of SGS sequences that match a query sequence, to identify genes and gene promoters with even with low coverage data [4].

TAGdb is a Web-based query tool which enables user to align query sequences to a collection of existing paired short reads. The TAGdb system was developed using Perl and MySQL and runs on a public web server (<http://sequencetagdb.info/tagdb/cgi-bin/index>). The interface allows researchers to upload or input a FASTA-formatted nucleotide sequence up to 5000 bp long and select one or more paired read sequence libraries from the TAGdb database for comparison. The input sequence is aligned to the selected library using MEGABLAST [5], and the resulting short reads with significant identity are visualized using a custom web interface. Each submitted job has a unique identifier, and an e-mail is sent to the user once the job has completed. The processing time for each search varies depending on the length of the input sequence and number of matching reads, but generally, searches are completed and results are returned within 20 s to 5 min. TAGdb hosts data for a variety of plant and fungal species and additional Illumina paired short-read sequence data may be hosted on request.

TAGdb is currently the only Web-based tool for searching short paired read sequence data. It is applicable to any species for which SGS data is available and particularly useful for rapid gene discovery in orphan or complex crop species based on homology to a gene from a model organism. Even in the case of poor data quality or high species divergence resulting in poor read coverage, user can employ the paired nature of SGS reads to design amplification primers to span the read pairs from TAGdb results for marker discovery or gene analysis.

2 Materials

With emerging sequencing technologies, supporting bioinformatics tools are useful to study sequence specificity of individual genome of various crops. In this view a wide range of whole-genome sequences of Brassica, Barley, Chickpea, Diplotaxis, Hirschfeldia, Lotus, Pongamia, Sinapis, Wheat, etc. are made available in TAGdb for searching the sequence of interest in respective cultivars. Additionally, on successful isolation and sequencing of Wheat group 7 chromosomes data (that has been used for syntenic builds [6]) and Chickpea chromosomes [7] (of desi and kabuli types) are also made available on TAGdb (Table 1).

Table 1
Summary of sequence data available on TAGdb

Data type	Volume of data (bp)	Coverage
Barley		
<i>Morex</i>	778,370,000	0.17×
Brassica		
<i>B. nigra</i>	1,110,000,000	1.76×
<i>B. oleracea</i>	1,160,000,000	1.67×
<i>B. rapa</i> (chiifu)	195,260,000	0.37×
<i>B. rapa</i> (kenshin)	427,090,000	0.81×
Chickpea		
Ambar	5,354,438,000	7.23×
Amethyst	3,569,681,400	4.82×
Barwon	4,373,273,400	5.90×
ICC4958	5,904,799,400	7.97×
Chromosome A ^{a,b}	1,862,661,400	20.1×
Chromosome B ^{a,b}	1,799,628,400	17.2×
Chromosome H ^{a,b}	1,657,472,400	32.6×
Almaz	4,298,195,700	5.80×
Bumper	4,882,263,300	6.59×
ICC8261	6,730,454,000	9.09×
Kaniva	4,630,218,600	6.25×
Chromosome A ^{a,c}	1,038,096,342	11.2×
Chromosome B ^{a,c}	805,459,000	11.2×
Chromosome C ^{a,c}	1,961,071,032	27.1×
Chromosome DE ^{a,c}	3,334,063,622	20.1×
Chromosome F ^{a,c}	1,638,068,670	12.5×
Chromosome G ^{a,c}	2,023,261,302	16.2×
Chromosome H ^{a,c}	257,184,408	5.05×
Diplotaxis		
<i>D. tenuifolia</i>	4,845,997,400	4.28×
Hirschfeldia		
<i>H. incana</i>	5,190,515,200	4.59×

(continued)

Table 1
(continued)

Data type	Volume of data (bp)	Coverage
Lotus		
AM3	5,147,454,800	10.2×
As2538	6,804,059,200	13.3×
DIMG-20	6,590,000,000	13.1×
Nicotiana		
<i>N. alata</i>	1,681,638,300	0.37×
Pongamia		
<i>P. pinnata</i>	450,042,048	0.26×
Sinapis		
<i>S. alba</i>	3,647,400,200	3.22×
Tetraselmis	2,779,882,820	185×
Wheat		
ACBarrie	36,930,989,000	2.17×
Alsen	11,578,205,400	1.36×
Baxter	38,855,331,600	2.28×
Chara	22,327,794,400	2.63×
Chinese spring		
Drysdale	14,915,336,000	0.88×
Excalibur	17,100,145,200	1.03×
Gladius	15,099,311,400	0.89×
H45	33,676,000,000	1.98×
Kukri	43,001,394,200	2.52×
Pastor	17,161,705,200	0.88×
RAC875	18,529,035,900	1.08×
VolcaniDDI	15,037,296,600	0.72×
Westonia	38,730,850,600	2.27×
Wyalkatchem	36,524,753,000	2.15×
Xiaoyan	18,252,000,000	1.07×
Yitpi	16,152,000,000	0.94×
Chromosome 7AS ^a	5,850,000,000	14.3×
Chromosome 7AL ^a	10,220,000,000	25.1×

(continued)

Table 1
(continued)

Data type	Volume of data (bp)	Coverage
Chromosome 7BS ^a	8,860,000,000	24.6×
Chromosome 7BL ^a	7,800,000,000	14.4×
Chromosome 7DS ^a	14,610,000,000	38.3×
Chromosome 7DL ^a	13,230,000,000	38.2×

^aIsolated chromosome sequence data^bDesi type (chickpea)^cKabuli type (chickpea)

3 Methods

To the researchers, TAGdb provides access to the genome sequence data being produced by new sequencing technologies. On identification of large number of pair reads matching may enable the local assembly of the genomic region. PCR amplifying the mapped read pairs allow to sequence the gene and genomic sequence flanking the matching region of query sequence. TAGdb (*see Note 1*) currently hosts whole-genome paired read libraries of Barley, Brassica, Diplotaxis, Hirschfeldia, Lotus, Nicotiana, Pongamia, Rya, Sinapis, Tetraselmis, and Wheat.

1. Choose the query sequence homologous to a gene from a model organism for aligning query sequences to an existing database of paired short-read data. In this example, we use *Triticum aestivum* FTD gene sequence.
2. First, identify the genes from National Centre for Biotechnology Information (NCBI) (<http://www.ncbi.nlm.nih.gov/>). Select “Nucleotide” database from the drop-down list and type “*Triticum aestivum* [orgn] AND FTD-h1” in the search box to find related sequences (*see Note 2*).
3. Select the hit (Accession: EF428113) from the results. Click on “send to” dropdown list to select “File” option as Destination and “FASTA” as format and click the “create File” button to download the sequences in FASTA format (*see Fig. 1*). Alternatively, select and copy sequence into clipboard by clicking on the link of selected NCBI search hit (*see Note 3*).
4. Open the TAGdb Web site (<http://sequencetagdb.info/tagdb/cgi-bin/index>) (*see Fig. 2*). Enter valid email address in provided text box and click on the “Choose File” to upload FASTA file or alternatively, paste FASTA format sequences in the provided text box (*see Note 4*). Select species from the list of species selection box to show the libraries available in library selection box. Select library(s) (*see Note 5*) and click on the “Start” button to start TAGdb pipeline.

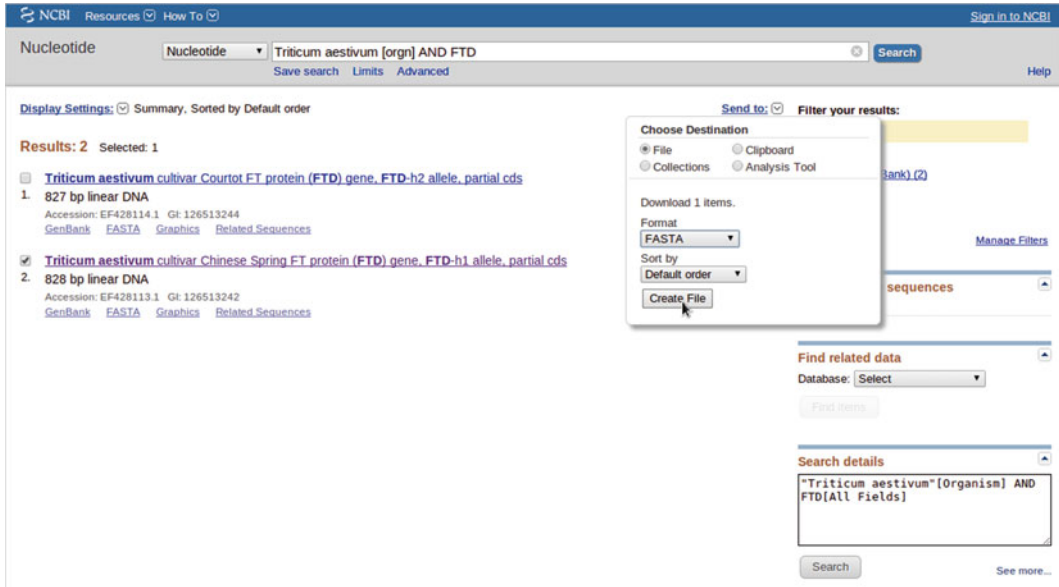


Fig. 1 Retrieval and downloading of wheat drought-related sequences from GenBank

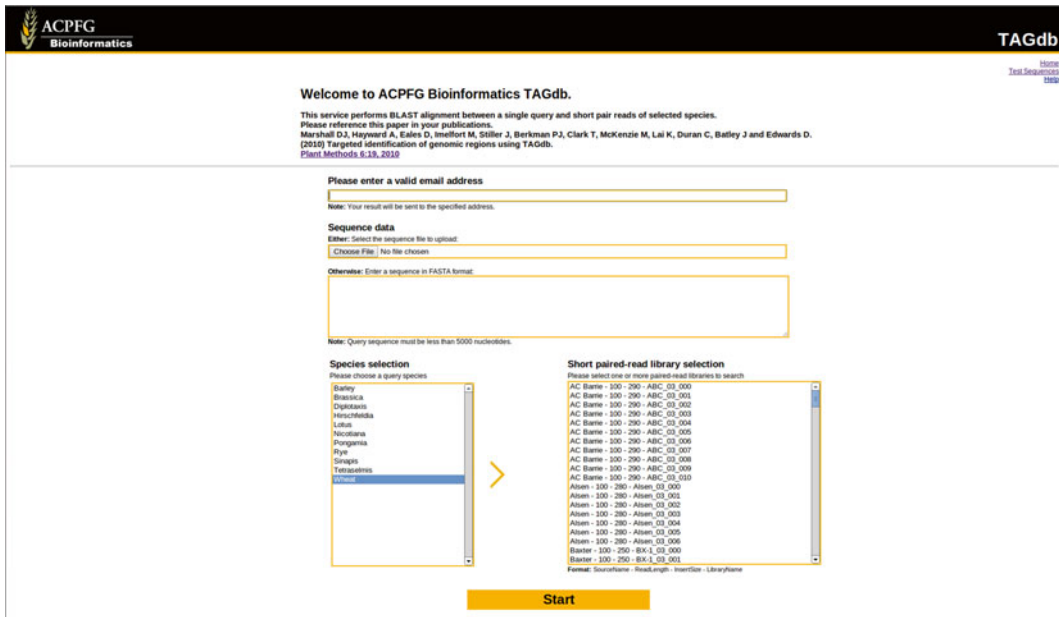


Fig. 2 Screenshot of TAGdb showing available species and paired read libraries of wheat

5. After starting the process, TAGdb sends an e-mail to the user stating that the job has started successfully and provides a link to the results web page.
6. Once search is complete, TAGdb send a second e-mail to confirm completion, together with a link to the results.

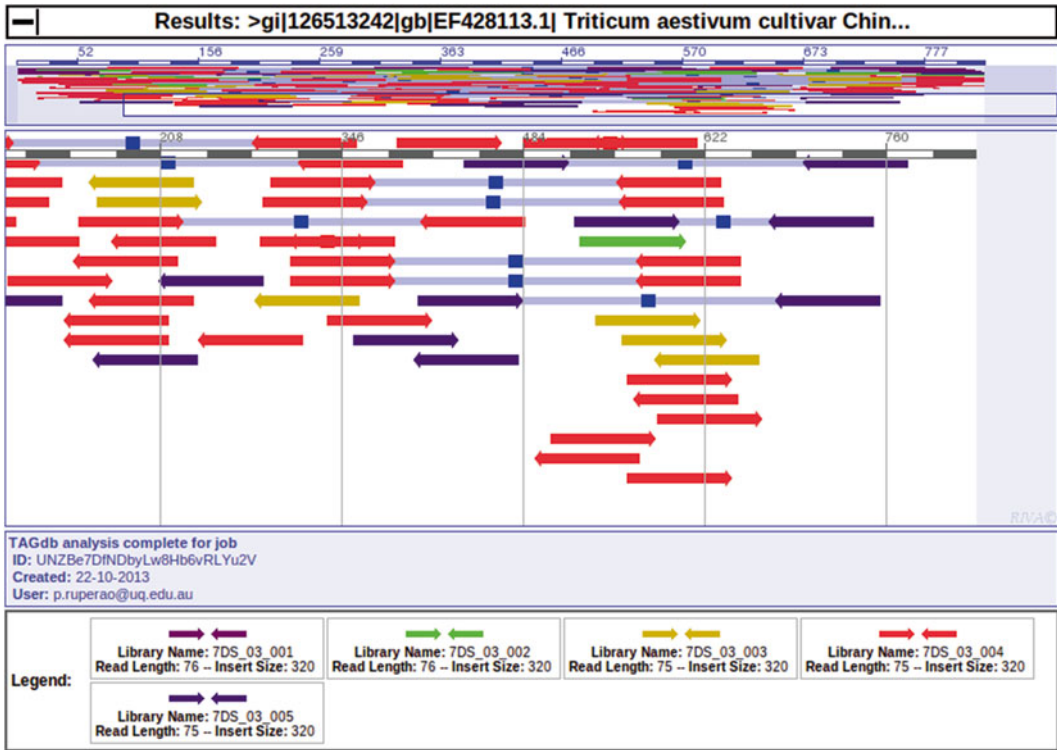


Fig. 3 Screenshot of TAGdb showing the alignment of short reads from *Triticum aestivum* 7DS. The input sequence is represented horizontally, with reads represented by arrows. Arrows are connected with a line if they represent paired reads

Tag alignment information (download fasta)						
ID	Start	End	Tag sequence	Orient	Library	Type
1	494	559	CCCTAAGGTTGGGATCGCTTGGACTTGGAGCATCTGGGCTACCATCCCTCGGGCAGTACACGGTGCACATCA	-1	7DS_03_001	P
1	306	381	AGCTCTGCTGGTGTGATGATGATTTCTATGCTGTTTTCTTTCTTTCG999GAACATGATTTTGGATGCTTCT	1	7DS_03_001	P
2	501	576	CACGCGTGTACTGCGCCGAGTGATGGTAGACCAGATGCCAAGTCCCAAGGATCCCAACCTTAGGGAGTATCT	1	7DS_03_001	P
2	737	812	GACAGCGTCTGCGCCGAGGCTGCTGGAGAACAGCAGCAGCAGGATGATCCCAATGGTGGAGGAGGCTC	-1	7DS_03_001	P
3	2	77	CAGGGTGACCTTCGGGAACAGGACCGTGTCCAACGGCTGGAGCTCAAGCCGTCCATGGTCCGCCAGCCCGG	1	7DS_03_001	P
3	241	316	ACCAGAGAGCTAGCAGAGGCCATATGCAAGATGGAGCATAGTCACTGCTGCCAAGACAACATCTGTGAAC	-1	7DS_03_001	P
4	580	655	CTGGTAAGTACTAAATTTGTAACTCAGTGAATAATTTCTGCTCCCTAGATACACAGTACAGTGTGTGTGT	1	7DS_03_001	P
4	744	819	CGGAGCGTACAGGCTCTGCGCCGAGCTGCTGAGAGAGCAGAGCAGGATGATCCATCCATGCTGGAGC	-1	7DS_03_001	P
5	687	772	CAACTGATGATCTTGGGAGAGAGGATGATGCTAGGAGAGCCCTCGCCAGATGAGGATCCATGCTGCTGT	1	7DS_03_001	A
5	-	-	TAATTAATAGTGGCTGGGTTTTATGAAAAAATATACTAGTAAATAGAAAAAGTTGGATAATTTGCTGAATTG	1	7DS_03_001	-
6	-5	70	TGCTGGGAGCATTGGAGCGCTTGGACTCGAGCCGTGGAGCAGGCTGCTGTTCCCAAGGTACCCGAGGCTGG	-1	7DS_03_001	A
6	-	-	AAGGAGTACTAGAGCGGCGAGCGGCTGAAGCTGCTGGACATGGACATGTACCCTGCTGAGCTTTTCGGTCC	-	7DS_03_001	-
7	588	663	ATGTAGACACACACATGACTGCTGTATATCTAGGAGCAGAGAAATATGCACTGAGTACAAATTTAGTA	-1	7DS_03_001	A
7	-	-	ACCATCACAGAGAGCCACCGGCTCTATACACAACCTTGAACCTTGTATATATGGCATAATCATAGGGGGCTTGT	1	7DS_03_001	-
8	-5	70	TGCTGGGAGCATTGGAGCGCTTGGACTCGAGCCGTGGAGCAGGCTGCTGTTCCCAAGGTACCCGAGGCTGG	-1	7DS_03_001	A
8	-	-	AAGGAGTACTAGAGCGGCGAGCGGCTGAAGCTGCTGGACATGGACATGTACCCTGCTGAGCTTTTCGGTCC	-	7DS_03_001	-
9	1376	1381	ATTCCTGATGAGCAGAGATGGAGATCACTGCTCCACAGCTCATATATAGAGGCTGATGATATGCTGGCTG	-1	7DS_03_001	A

Fig. 4 List of TAG sequences showing significant similarity with query sequence. Table showing TAG start, end positions, orientations, library, and type of TAG sequence

- The result web page consists of two windows displaying an overview and zoomed region of the read alignments (see Figs. 3 and 4). Paired reads are connected by a line, with a blue rectangle confirming that the result conforms to the expected orientation and paired read distance.
- Matching reads, together with their matching or non-matching read pairs, are viewed as a table or can be downloaded as a multi-FASTA format file for further analysis.

4 Notes

1. TAGdb is a pipeline developed in perl by integrating with MEGABLAST and MySQL.
2. Currently NCBI search produces only one sequence for FTD-h1 in *Triticum aestivum*, but the number may increase as the database size increases.
3. Alternatively, these sequences may be downloaded from http://sequencetagdb.info/tagdb/cgi-bin/download?key=testseq_Wheat_7DS_gene.
4. TAGdb request a FASTA format query sequence of up to 5000 bp length.
5. To select multiple paired read libraries, press “Ctrl” key and select libraries.

References

1. Kircher M, Kelso J (2010) High-throughput DNA sequencing—concepts and limitations. *Bioessays* 32(6):524–536
2. Batley J, Edwards D (2009) Genome sequence data: management, storage, and visualization. *Biotechniques* 46(5):333–334
3. Lee HC et al (2012) Bioinformatics tools and databases for analysis of next-generation sequence data. *Brief Funct Genomics* 11(1):12–24
4. Marshall DJ et al (2010) Targeted identification of genomic regions using TAGdb. *Plant Methods* 6:19
5. Zhang Z et al (2000) A greedy algorithm for aligning DNA sequences. *J Comput Biol* 7(1–2):203–214
6. Berkman PJ et al (2011) Sequencing and assembly of low copy and genic regions of isolated *Triticum aestivum* chromosome arm 7DS. *Plant Biotechnol J* 9(7):768–775
7. Ruperao P et al (2014) A chromosomal genomics approach to assess and validate the desi and kabuli draft chickpea genome assemblies. *Plant Biotechnol J* 12(6):778–786

Short Read Alignment Using SOAP2

Bhavna Hurgobin

Abstract

Next-generation sequencing (NGS) technologies have rapidly evolved in the last 5 years, leading to the generation of millions of short reads in a single run. Consequently, various sequence alignment algorithms have been developed to compare these reads to an appropriate reference in order to perform important downstream analysis. SOAP2 from the SOAP series is one of the most commonly used alignment programs to handle NGS data, and it efficiently does so using low computer memory usage and fast alignment speed. This chapter describes the protocol used to align short reads to a reference genome using SOAP2, and highlights the significance of using the in-built command-line options to tune the behavior of the algorithm according to the inputs and the desired results.

Key words Next-generation sequencing, Short read alignment, Read mapping, Gapped alignment, Ungapped alignment, Burrows–Wheeler transform (BWT), Nucleotides, Mismatches, Repeats, Match mode, Seed length, Genomeindexing, SNPprediction, Genomics, Structural variant

1 Introduction

The advent of next-generation sequencing (NGS) technologies has given rise to a vast amount of genomic data for numerous applications in life sciences. These include metagenomics [1], SNPprediction [2] genomic structural variant detection, DNA methylation studies [3], mRNA expression analysis [4], cancer genomics [5], and personalized medicine [6]. The first step prior to any kind of downstream analysis involves aligning DNA sequence fragments in the form of sequenced reads to an appropriate reference genome. The aim of sequence alignment is to find the location within the reference genome that corresponds best to the observed DNA sequence. The regions of similarity can be due to functional, structural, or evolutionary relationships between the sequences [7].

The process of aligning reads to a reference is a multistep approach [8]. In the first step, heuristic techniques such as the hash table approach or the Burrows–Wheeler Transform are used to

locate a small set of places in the reference genome which also contains the region that best corresponds to the read. When all the possible places have been identified, slower and more accurate alignment algorithms such as Smith-Waterman are run on this subset of places to determine the best location. Short read alignment is computationally challenging, which explains why the most widely used alignment programs for NGS data have been specifically designed to address this issue.

1.1 Hash Table Approach and the Burrows–Wheeler Transform (BWT) Approach

The hash table method makes use of the hash table data structure, which is able to index data so that rapid searching can be performed [9]. This is convenient for DNA sequencing data, which usually contain duplicate reads, but unlikely to contain every possible combination of the four nucleotides “A”, “C”, “T” and “G”. Hash-based algorithms require that a hash table is built on either the set of reads or the reference genome. The reference genome is then used to scan the table for reads or vice versa. The memory requirement for the hash table method differs depending on whether the data structure was built on the reference genome or the input reads. In the former case, the amount of memory will be constant for a given set of parameters, and will be independent of the number of reads used, although this may vary based on how complex and big the genome is. However, in the latter case, the amount of memory needed will be small and variable depending on the number and diversity of the input reads such that more processing time will be required to scan the entire reference genome when there are relatively few reads [8].

More recent alignment algorithms have been designed based on the BWT approach, which uses the FM index (Full-text index in Minute space) [10] data structure. According to this approach, when a suffix array is built using the BWT sequence instead of the original sequence, it becomes more efficient. A suffix array consists of the start positions of suffixes of the genome, which have been sorted in a particular (lexicographic) order [11]. There are two steps involved when creating the FM index. In the first step, BWT is used to reorder the sequence of the reference genome such that sequences that exist multiple times occur together in the data structure. Next, when the final index is created, it is used to match reads to the genome. Similar to hash-based methods, once the reads have been associated with the region of the genome they are more likely to align to, more sensitive algorithms can be used to complete the alignment process [8].

Both BWT-based and hash table-based methods have their advantages and disadvantages. BWT algorithms have been reported to be much faster than hash table-based approaches, even in situations where a read can map to multiple locations in the reference genome. This is because the reference is somehow “collapsed” such that repeated regions are scanned only once [11, 12].

The other advantage of BWT-based approaches is that the complete reference genome index can be stored on disk and loaded completely into memory on almost all standard bioinformatics computing clusters, but there will be a trade-off between speed and sensitivity [9]. However, error handling is not the forte of BWT algorithms, and increasing the number of errors increases the computational time required for mapping. Hash table-based methods on the other hand appear to better handle errors, provided that the errors are not uniformly spread across the read. In addition, seeds of highly repetitive regions are not specific, and therefore mapping of such regions can be very slow [11].

1.2 Short Oligonucleotide Analysis Package (SOAP)

There are numerous NGS alignment tools available today, each with their own specificities. Therefore, which one to use for a given application can be tricky. There are a number of criteria that can influence the choice of the mapping tool to be used: the type of algorithm that is used (hash-based or BWT), whether it can use multiple processors at the same time, the type of read (paired-end or single-end) that it can process, its ability to allow gapped/ungapped alignments, and the output format [13].

One of the most commonly used short read aligners for NGS data is SOAP. The first version, which was implemented in 2008 [14] uses the hash table approach to align reads to a reference genome. It was specifically designed to detect and genotype SNPs via ungapped alignment, while also supporting gapped alignments. In addition, it provides special features, such as the alignment of paired-end reads, small RNA and mRNA sequence tags.

The second version of SOAP, i.e., SOAP2 [15] provides the same features as the previous one but greatly improves on it by making use of the BWT approach to index the reference sequence. This reduces the time and memory required for the alignment of reads. With longer reads now being produced by sequencing machines, SOAP2 can process reads of up to 1024 bp in length. The aligner uses a “split-read strategy” when performing gapped and ungapped alignments simultaneously. To allow one mismatch, the read is split into two fragments. The mismatch can exist in, at most, one of the two fragments at the same time. In a similar way, a read is split into three fragments to search for hits that allow two mismatches. This approach is used to identify mutation sites on the reads. Apart from read alignment, SOAP2 offers additional features including assembly of the aligned reads into a consensus sequence which can be used for SNP detection by comparing the assembled genome to the reference.

Yet another version of SOAP came out in 2013. This version called SOAP3 [16] uses the same BWT approach as SOAP2, but exploits the multi-processing capacity of a graphical processing unit (GPU) to considerably reduce the time taken for alignment. As a result, it can align reads up to ten times faster than its

predecessor. This is particularly important in the current era where longer reads and larger volumes of data are being produced. However, SOAP3 has a larger index and needs more time to load it, but this is circumvented by its high alignment speed. This version supports ungapped alignment only, allowing up to four mismatches, but SOAP3-dp [17], which is an improved version of SOAP3, allows both ungapped and gapped alignments.

The rest of this chapter focuses on SOAP2 as it is the most commonly used version in the SOAP series.

2 Case Study

This section demonstrates how to use SOAP2 to align short reads to an appropriate reference genome. Choosing which flag to use for read mapping can be quite a daunting task, and will be highly influenced by the type of downstream analysis to be performed. SOAP2 offers a multitude of these features, and while their use is described in the manual (Table 1), one might still find it tricky to decide when and how to use some of these features, especially if the user is new to read mapping. This case study will therefore give some advice regarding how these options should be set to suit the desired application.

The dataset used for experimentation was derived from the data generated by Minoche et al. [18] in a study to evaluate NGS data from Illumina HiSeq2000. The entire dataset consists of 71 million read pairs (100 bp long reads) containing 99 % genomic DNA of *Arabidopsis thaliana* and 1 % bacteriophage PhiX174. After quality evaluation of the raw reads, it was found that 53,540,169 read pairs which constitute 74 % of all sequenced bases, had a Phred quality score of at least 30. This subset of reads, which can be found in the Sequence Read Archive (accession number SRX101463), was used for the purpose of analysis.

Since the reads had already been selected based on quality score, base trimming was not performed, and the entire read length was used for alignment. The reference used for read mapping was the complete *A. thaliana* genome sequence (Athaliana_167), which can be found on Phytozome v9.1 [19].

The first step in any read mapping experiment with SOAP2 consists of creating the Burrows–Wheeler index of the reference genome using the “2bwt-builder” command. This is done as shown below:

```
$ 2bwt-builder Athaliana_167.fa
```

This command generates a total of 13 index files from the reference genome FASTA file as follows:

```
Athaliana_167.fa.index.amb, Athaliana_167.fa.index.ann,
Athaliana_167.fa.index.bwt, Athaliana_167.fa.index.fmv,
Athaliana_167.fa.index.hot, Athaliana_167.fa.index.lkt,
```

Table 1
Options offered by SOAP2 [15]

Parameter	Description
-a < str>	Query a file, *.fq, *.fa
-b < str>	Query b file
-D < str>	Reference sequences indexing table, *.index format
-o < str>	Output alignment file (txt)
-M < int>	Match mode for each read or the seed part of read, which should not contain more than two mismatches, [4]
	0: exact match only
	1: 1 mismatch match only
	2: 2 mismatch match only
	4: find the best hits
-u < str>	Output unmapped reads file
-t	Output reads ID instead reads name, [none]
-l < int>	Align the initial n bps as a seed [256] means whole length of read
-n < int>	Filter low-quality reads containing >n Ns before alignment, [5]
-r [0,1,2]	How to report repeat hits, 0=none; 1=random one; 2=all, [1]
-m < int>	Minimal insert size allowed, [400]
-x < int>	Maximal insert size allowed, [600]
-2 < str>	Output file of unpaired alignment hits
-v < int>	Maximum number of mismatches allowed on a read. [5] bp
-s < int>	Minimal alignment length (for soft clip) [255] bp
-g < int>	One continuous gap size allowed on a read. [0] bp
-R	for long insert size of pair end reads RF. [none] (means FR pair)
-e < int>	will not allow gap exist inside n-bp edge of a read, default=5
-p < int>	Number of processors to use, [1]
-h	This help

Athaliana_167.fa.index.pac, Athaliana_167.fa.index.rev.bwt,
 Athaliana_167.fa.index.rev.fmv, Athaliana_167.fa.index.rev.lkt,
 Athaliana_167.fa.index.rev.pac, Athaliana_167.fa.index.sa,
 Athaliana_167.fa.index.sai.

These can be used across multiple runs of SOAP2. Once these index files are generated, read alignment can be performed.

Running SOAP2 is fairly straightforward. The algorithm, which operates in the Linux environment, essentially makes use of command-line arguments that allow the user to tune its behavior according to the inputs and the desired result. The user simply types “soap”, followed by a combination of the parameters displayed in the Table 1 above. An example of a SOAP2 command is shown below:

```
$ soap -p 2 -l 50 -r 0 -v 2 -m 1 -x 1000 -a forward_read.fastq
-b reverse_read.fastq -D Athaliana_167.fastq.index -o reads.
mapped -2 reads.unpaired -u reads.unmapped
```

Any SOAP2 run would result in three output files with the following extensions: (1) .mapped, (2) .unmapped, (3) .unpaired. The .mapped and .unpaired files consist of the paired-end and single-end reads, respectively, that have mapped to the reference, while the .unmapped file contains reads that did not map to the reference. Both the mapped and unpaired reads are reported in the SOAP format, which is tab-delimited and very similar to the SAM (Sequence Alignment/Map) format. The first line of the SOAP output obtained for one of the read mappings (*-r 0 -v 2*) is represented in Table 2.

Each column representing one aspect of the mapping result as explained below:

1. ID of the read
2. Full sequence of the read; the read will be reverse-complemented if mapped to the reverse chain of the reference.
3. Quality of the sequence read; this will be reverse-complemented too if the read is mapped to the reverse chain of the reference.
4. Number of equal best hits; reads with no hits will be reported as unmapped.
5. This flag can either be “a” or “b” and is only reported when mapping paired-end reads to the reference; “a” corresponds to the forward read while “b” corresponds to the reverse read.
6. Length of the read; if the read is trimmed, then the new trimmed length is shown.
7. This flag can be either “+” or “-” to indicate if the read mapped to the forward (+) or reverse (-) chain of the reference.
8. ID or chromosome number of the reference sequence
9. Location in base pairs of the reference where the read was mapped.
10. Type of hit reported; “0” refers to an exact match while “1” or “2” relate to the number of mismatches, followed by the mutation and the location on the reference where the mutation occurred. In this example, there is only one mismatch between the reference and the read, and this occurs on the 24th bp of the reference from position number 9931404, and

Table 2
SOAP output for one of the read mappings

Column number	Contents
1	SRR353664.30.1
2	TGCTTCCACGACCATTGGAGCCGCCGACCACGAGTAGAAGGCTGGTTTTCTTGGGGATTTTGCTCTAACCCAGCCAGGATATC CATGAACCAGAAAGCA
3	#####@DD0GCFEDFFB:EEEEEEEEFF@CHFFHHHHHHHHFHGHGEDFDDDDDDDD#EEEEHHGHHHHGH HHHHHHHHHHHHHH
4	1
5	a
6	100
7	-
8	Chr2
9	9931404
10	1
11	T->24G6 100M
12	74T25

the mismatch is a “T” on the reference as opposed to a “G” on the read; the number “6” refers to the quality of “C” allele in the read.

11. This column forms part of the CIGAR (Compact Idiosyncratic Gapped Alignment Report) string [20]. The letter “M” is used to indicate that a match/mismatch has occurred. If gapped alignments are performed, then there may also be an “I” or a “D” if an insertion or a deletion has occurred, respectively.
12. This column is also part of the CIGAR string. The string in this case is “74T25”, and indicates that (1) the first 74 bases are the same on both the reference and the read, (2) a base substitution occurred at position 75 on reference where a “T” was swapped for a “G”, and (3) the remaining 25 bases are the same between the reference and the read. If no mismatches are recorded, then this string will only show the number of bp that constitute the read length.

3 Algorithmic Features of SOAP2

Usually, using a mapping tool’s default parameter values would lead to good quality output. When `-r 1 -v 5` (default options in SOAP2) are used, 70.5 % of reads are aligned to the *A. thaliana* reference. However, there are situations where these parameters and additional ones such as `-l`, `-M`, and `-g` may have to be tweaked to suit the type of downstream analysis to be performed. Therefore, it is important to understand the effect of using them, as well as the compromises that are made when using them.

3.1 Number of Mismatches (-v)

In cases where sufficiently close reference genomes are not available [21], more relaxed mappings should be performed by allowing more mismatches. SOAP2, by default, allows five mismatches in the read, but this value can be adjusted as per the user’s requirements. For instance, stringent mappings may be required in which case the lowest number of mismatches that the algorithm of choice can possibly allow should be used. This would be the same as using `-v 2` in SOAP2.

3.2 Seed Length (-l)

While the mapping percentage increases nonlinearly with the number of mismatches [13], it is important to note that there is a correlation between the number of mapped reads and the combination of seed length and number of mismatches allowed during read mapping. In SOAP2, if the `-l` parameter is not specified in the command, then the entire read length is used as the seed by default. The algorithm allows a maximum of 2 mismatches in the seed portion of the read, which corresponds to the 5’-end, or the

high-quality part of the read [7, 13]. Increasing the length of the seed also means reducing the size of the non-seed portion of the read, and vice versa. The non-seed portion of the read includes the 3'-end, which is the low quality region of the read. Since this region contains more sequencing errors, more mismatches should be allowed here to increase read mapping [7, 15]. Figure 1 which follows shows how varying the seed length for different values of $-v$ impacts on the percentage of mapped reads.

Using a seed during read alignment increases performance and accuracy [13]. The general trend shows an increase in the percentage of mapped reads for increasing values of $-l$ and $-v$, with the highest percentage of mapped reads achieved with $-v$ 5. Also, when the entire read is used as the seed, the value of $-v$ does not seem to affect the percentage of mapped reads.

3.3 Match Mode (-M)

The $-M$ parameter also appears to play a role in the stringency of read mappings. While allowing the least number of mismatches is usually preferable, using $-M$ 2 will also significantly reduce the number of mapped reads (Fig. 2). It also appears that when the $-M$ flag is omitted from the SOAP command, the algorithm will find the best match possible. This is the same as using $-M$ 4.

3.4 Reporting Repeat Hits (-r)

Mapping reads to highly repetitive genomes can be very ambiguous and can lead to erroneous interpretation of results. Repetitive DNA is present in all kingdoms of life, and in particularly high levels in plant genomes. The *A. thaliana* genome, for instance comprises repeats in the form of large segmental duplications [22]. Read mapping to repetitive genomes is further exacerbated when reads align to multiple locations on the reference. Such multi-reads can affect downstream analyses such as SNP calling where false positives and false negatives may occur [23].

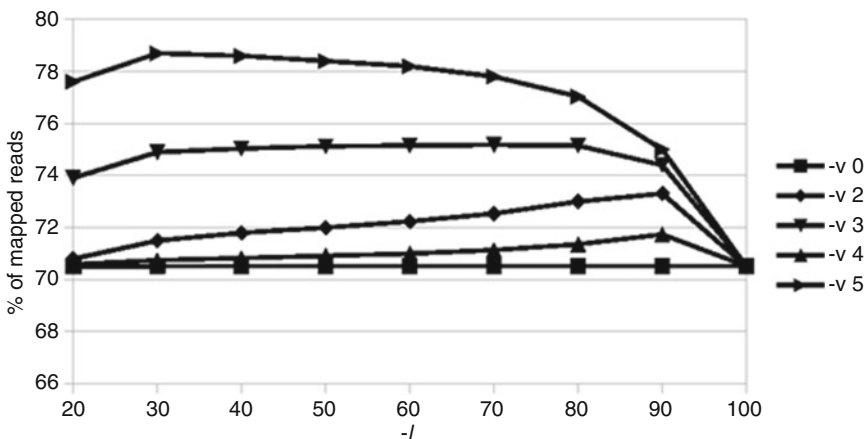


Fig. 1 Effect of varying seed length ($-l$) and number of mismatches ($-v$) on read mapping ($-r$ 1 and $-M$ 4)

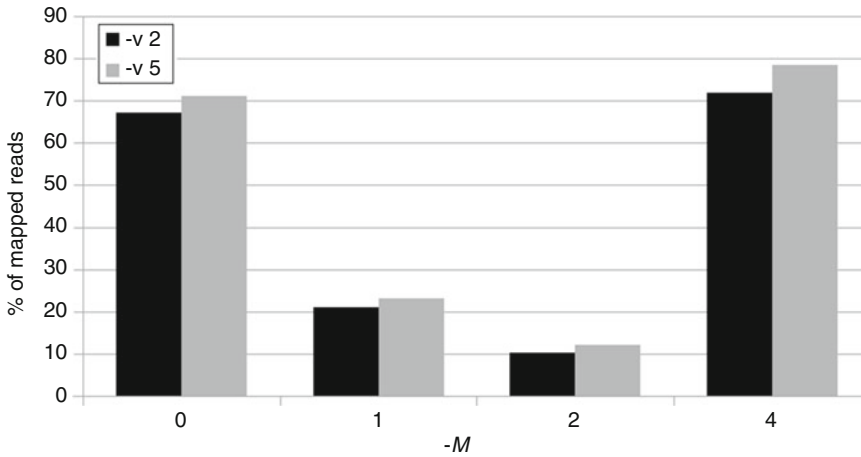


Fig. 2 Effect of varying the value of the match mode ($-M$) when allowing different numbers of mismatches ($-v$) for $-r 1$ and $-r 50$

SOAP2 deals with multi-reads in three ways. The first one is to discard all multi-reads. This translates as using $-r 0$. This approach was used by Lorenc et al. [24] to call SNPs between four different hexaploid wheat cultivars across chromosomes 7A, 7B and 7D. Apart from removing reads that map to multiple locations with the same affinity, the $-r 0$ option also discarded reads that could not be accurately positioned on the reference, thereby increasing the accuracy of SNP calling. In this way they were able to discover more than 800,000 SNPs with a validation rate greater than 93 %.

The second way to deal with multi-reads is the “best match” approach, where the alignment with the lowest number of mismatches is reported. This equates to using $-r 1$. In cases where more than one equally best match is found, one will be chosen at random and reported. The third and last option is to report all alignments that are found. This can be done by using $-r 2$. This is particularly useful in applications such as copy number variant (CNV) calling [25].

Depending on the nature of the dataset, $-r 1$ and $-r 2$ may result in the same percentage of mapped reads, which would suggest that some reads have exactly one best match in the reference genome. Although this may be the case here (Fig. 3), it does not always happen.

3.5 Gapped Alignment ($-g$)

Gapped alignment is another important feature that SOAP2 offers for paired-end reads via the $-g$ flag. Correct alignments between real, related sequences may contain gaps, which could be due to insertion/deletion (indel) polymorphisms [13, 26]. Whether or not to allow gaps during read mapping will depend largely on the dataset. However, it is expected that the number of mapped reads

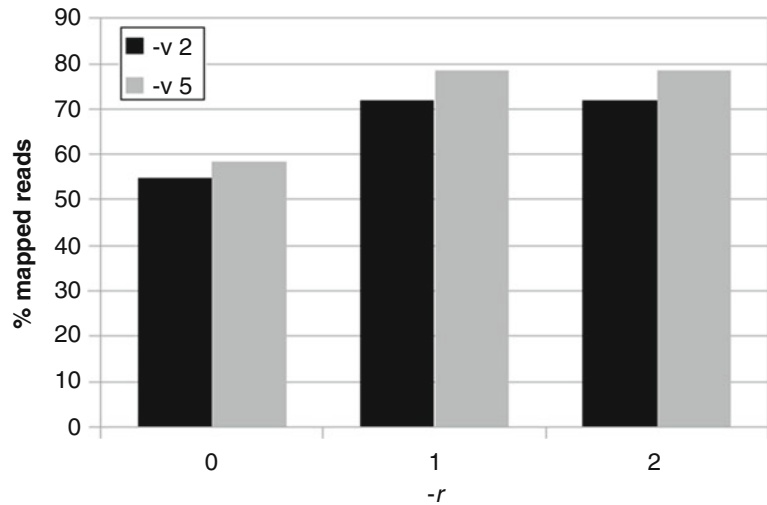


Fig. 3 Effect of reporting repeat hits ($-r$) on the number of mapped reads for different number of mismatches allowed ($-v 2$ and $-v 5$)

will increase when gapped alignments are performed. SOAP2 by default performs ungapped alignments [13] unless the $-g$ option is specified by the user in which case a continuous gap of up to 3 bp is permitted in one read only, while the other read should match its target exactly [7]. When comparing any two sequences, a SNP is more likely to occur than an indel, so the algorithm considers a mismatch to be less costly than a gap.

References

1. Qin J, Li R, Raes J, Arumugam M, Burgdorf KS, Manichanh C, Nielsen T, Pons N, Levenez F, Yamada T, Mende DR, Li J, Xu J, Li S, Li D, Cao J, Wang B, Liang H, Zheng H, Xie Y, Tap J, Lepage P, Bertalan M, Batto JM, Hansen T, Le Paslier D, Linneberg A, Nielsen HB, Pelletier E, Renault P, Sicheritz-Ponten T, Turner K, Zhu H, Yu C, Li S, Jian M, Zhou Y, Li Y, Zhang X, Li S, Qin N, Yang H, Wang J, Brunak S, Dore J, Guarner F, Kristiansen K, Pedersen O, Parkhill J, Weissenbach J, Bork P, Ehrlich SD, Wang J (2010) A human gut microbial gene catalogue established by metagenomic sequencing. *Nature* 464(7285):59–65
2. Van Tassell CP, Smith TP, Matukumalli LK, Taylor JF, Schnabel RD, Lawley CT, Haudenschild CD, Moore SS, Warren WC, Sonstegard TS (2008) SNP discovery and allele frequency estimation by deep sequencing of reduced representation libraries. *Nat Methods* 5(3):247–252
3. Taylor KH, Kramer RS, Davis JW, Guo J, Duff DJ, Xu D, Caldwell CW, Shi H (2007) Ultradeep bisulfite sequencing analysis of DNA methylation patterns in multiple gene promoters by 454 sequencing. *Cancer Res* 67(18):8511–8518
4. Sultan M, Schulz MH, Richard H, Magen A, Klingenhoff A, Scherf M, Seifert M, Borodina T, Soldatov A, Parkhomchuk D, Schmidt D, O’Keeffe S, Haas S, Vingron M, Lehrach H, Yaspo ML (2008) A global view of gene activity and alternative splicing by deep sequencing of the human transcriptome. *Science* 321(5891):956–960
5. Guffanti A, Iacono M, Pelucchi P, Kim N, Solda G, Croft LJ, Taft RJ, Rizzi E, Askarian-Amiri M, Bonnafant RJ, Callari M, Mignone F, Pesole G, Bertalot G, Bernardi LR, Albertini A, Lee C, Mattick JS, Zucchi I, De Bellis G (2009) A transcriptional sketch of a primary human breast cancer by 454 deep sequencing. *BMC Genomics* 10:163
6. Auffray C, Chen Z, Hood L (2009) Systems medicine: the future of medical genomics and healthcare. *Genome Med* 1(1):2

7. Yu X, Guda K, Willis J, Veigl M, Wang Z, Markowitz S, Adams MD, Sun S (2012) How do alignment programs perform on sequencing data with varying qualities and from repetitive regions? *BioData Min* 5(1):6
8. Flicek P, Birney E (2009) Sense from sequence reads: methods for alignment and assembly. *Nat Methods* 6(11 Suppl):S6–S12
9. Flicek P (2009) The need for speed. *Genome Biol* 10(3):212
10. Ferragina P, Manzini G (2005) Indexing compressed text. *J ACM* 52(4):552–581
11. Schbath S, Martin V, Zytnicki M, Fayolle J, Loux V, Gibrat JF (2012) Mapping reads on a genomic sequence: an algorithmic overview and a practical comparative analysis. *J Comput Biol* 19(6):796–813
12. Ruffalo M, LaFramboise T, Koyuturk M (2011) Comparative analysis of algorithms for next-generation sequencing read alignment. *Bioinformatics* 27(20):2790–2796
13. Hatem A, Bozdogan D, Toland AE, Catalyurek UV (2013) Benchmarking short sequence mapping tools. *BMC Bioinformatics* 14:184
14. Li R, Li Y, Kristiansen K, Wang J (2008) SOAP: short oligonucleotide alignment program. *Bioinformatics* 24(5):713–714
15. Li R, Yu C, Li Y, Lam TW, Yiu SM, Kristiansen K, Wang J (2009) SOAP2: an improved ultra-fast tool for short read alignment. *Bioinformatics* 25(15):1966–1967
16. Liu CM, Wong T, Wu E, Luo R, Yiu SM, Li Y, Wang B, Yu C, Chu X, Zhao K, Li R, Lam TW (2012) SOAP3: ultra-fast GPU-based parallel alignment tool for short reads. *Bioinformatics* 28(6):878–879
17. Luo R, Wong T, Zhu J, Liu CM, Zhu X, Wu E, Lee LK, Lin H, Zhu W, Cheung DW, Ting HF, Yiu SM, Peng S, Yu C, Li Y, Li R, Lam TW (2013) SOAP3-dp: fast, accurate and sensitive GPU-based short read aligner. *PLoS One* 8(5), e65632
18. Minoche AE, Dohm JC, Himmelbauer H (2011) Evaluation of genomic high-throughput sequencing data generated on Illumina HiSeq and genome analyzer systems. *Genome Biol* 12(11):R112
19. Goodstein DM, Shu S, Howson R, Neupane R, Hayes RD, Fazo J, Mitros T, Dirks W, Hellsten U, Putnam N (2012) Phytozome: a comparative platform for green plant genomics. *Nucleic Acids Res* 40(D1):D1178–D1186
20. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R (2009) The Sequence Alignment/Map format and SAMtools. *Bioinformatics* 25(16):2078–2079
21. Reynoso V, Putonti C (2011) Mapping short sequencing reads to distant relatives. In: *Proceedings of the 2nd ACM conference on bioinformatics, computational biology and biomedicine, 2011*. ACM, Chicago, IL, p 420–424
22. Arabidopsis Genome Initiative (2000) Analysis of the genome sequence of the flowering plant *Arabidopsis thaliana*. *Nature* 408(6814):796–815
23. Li H, Homer N (2010) A survey of sequence alignment algorithms for next-generation sequencing. *Brief Bioinform* 11(5):473–483
24. Lorenc MT, Hayashi S, Stiller J, Lee H, Manoli S, Ruperao P, Visendi P, Berkman PJ, Lai K, Batley J, Edwards D (2012) Discovery of single nucleotide polymorphisms in complex genomes using SGSautoSNP. *Biology* 1(2):370–382
25. Siragusa E, Weese D, Reinert K (2013) Fast and accurate read mapping with approximate seeds and multiple backtracking. *Nucleic Acids Res* 41(7), e78
26. Mott R, Tribe R (1999) Approximate statistics of gapped alignments. *J Comput Biol* 6(1):91–112

Chapter 14

Tablet: Visualizing Next-Generation Sequence Assemblies and Mappings

Iain Milne, Micha Bayer, Gordon Stephen, Linda Cardle,
and David Marshall

Abstract

This chapter is designed to be a practical guide to using Tablet for the visualization of next/second-generation (NGS) sequencing data. NGS data is being produced more frequently and in greater data volumes every year. As such, it is increasingly important to have tools which enable biologists and bioinformaticians to understand and gain key insights into their data. Visualization can play a key role in the exploration of such data as well as aid in the visual validation of sequence assemblies and features such as single nucleotide polymorphisms (SNPs). We aim to show several use cases which demonstrate Tablet's ability to visually highlight various situations of interest which can arise in NGS data.

Key words Visualization, Mapping, Assembly, Next-generation sequencing, SNPdiscovery

1 Introduction

The development of next-generation sequencing (NGS) technologies has had far-reaching effects on both the fields of genetics and bioinformatics. NGS has provided a cheap, powerful, and popular approach to sequencing whole genomes quickly and in turn has pushed bioinformatics researchers and programmers to create new software for analyzing the huge volumes of sequence data such techniques can provide. The volume of data that can be yielded from a sequencing run is still increasing, with ramifications for the selection of computing hardware used in bioinformatics infrastructure. For NGS data analysis, efficient software which maximizes use of disk space and computer memory is now essential. Tablet [1, 2] is optimized for high-volume assembly data, even from highly fragmented, unfinished genomes, and is designed to provide researchers with valuable insights into the results of a sequence assembly experiment. This chapter will take the user through five

different commonly occurring research scenarios and describe how Tablet's features can be used to assist the assessment and analysis of NGS sequence assembly data.

2 Materials

The methods described in this chapter rely upon the use of Tablet, a lightweight, high-performance and memory-efficient viewer for NGS assembly and alignment data. It is available for Windows, OS X, and Linux in both 32- and 64-bit versions. Tablet allows users to easily visualize their read data against reference data, including scaled-to-fit and coverage overviews and an extremely flexible main display allowing for visualization of the data at many different resolutions. Tablet supports all major assembly formats including the now de-facto standard SAM/BAM format [3], as well as the GFF3, BED, and VCF formats for importing of supplementary annotation data. Support is included for both single-end and paired-end visualization where appropriate, in both stacked (one read/pair per line) and packed (reads/pairs packed as close to the top of the display as possible) viewing modes, as well as multiple color schemes to depict features such as variants, read direction, read group, and so on. Tablet is distributed as a self-contained installable executable (with no other external dependencies) and offers periodic automatic update functionality to ensure that users are always running the latest version. Tablet can be downloaded from <http://ics.hutton.ac.uk/tablet>.

2.1 User Interface

Tablet's user interface (UI) is comprised of many separate components (*see* Fig. 1). We will refer to these components often throughout the rest of this chapter, but firstly we'll take a look at how to get data loaded into Tablet.

2.2 Loading Primary Data

Although Tablet supports many file formats, there are only three types of file that can be imported into it: files generated by de novo assembly or read mapping; reference (or consensus) files associated with a mapping; and annotation files that dictate points of interest along a reference. For example, you may have an assembly in BAM format, a reference in FASTA format, and an annotation in GFF3 format. It is possible to load just an assembly (e.g., ACE or AFG format), in which case no separate reference sequence is required, or a mapping file on its own (e.g., BAM or SAM format); in the latter case however, a reference sequence is also required if you wish to compare the mapping directly against the reference.

1. Click on the *OpenAssembly* button located in the *Home|Data* section of the toolbar.
2. Browse for and select the assembly or mapping file to be opened.

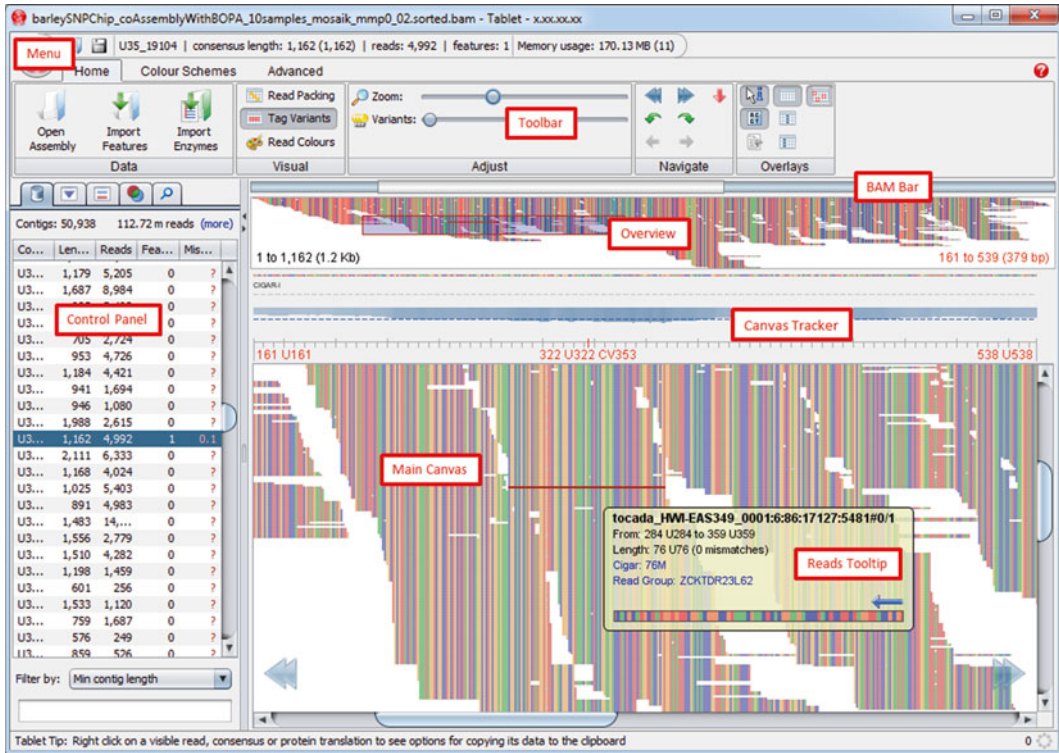


Fig. 1 Typical view of Tablet's graphical user interface, showing the terminology used for the main components present in Tablet

3. If required, select the reference file to be opened. This will be in FASTA format. Tablet will only enable this option if it detects a mapping file that requires an associated reference (e.g., the ACE format embeds assembly and reference together into a single file, but mappings in BAM/SAM format require a separate reference sequence).
4. Finally, click *Open* to import the data into Tablet (*see Note 1*).

Once the data has finished loading, you will see a screen which invites you to “select a contig to begin visualization”. Do this by selecting a contig from the contigs list on the left hand side of the screen. Click any row to load and display the reads and reference sequence that are associated with that contig.

Additionally Tablet supports the loading of annotation data in the form of GFF files. To load an annotation file:

1. Click the *Import Features* button from the *Home|Data* section of the toolbar to open a file browser.
2. Browse for your file, select it, and click open. Tablet will import your features and the first three feature types from the file will be displayed on the feature track, which can be found below the reference sequence, but above the reads.

3. The visible tracks can be edited by clicking on the *Select Tracks* link located within the *Features* tab of the control panel, or by right-clicking on any of the actual tracks and picking the option from the pop-up menu (see **Note 2**).

2.3 Using Tablet

In this section, we will briefly describe some of the core features available within Tablet, and describe operations that are essential to working through the examples presented later in this chapter.

1. The reads forming an assembly/mapping are presented on the main canvas. It is possible to navigate around this area using the scrollbars or by clicking and dragging with the mouse on the canvas itself. The canvas only shows a subset of the overall data (dependent on the current zoom level)—the overview canvas summarizes the current contig and the viewing position of the main canvas, which will be highlighted with a red rectangle.
2. Detailed read information is available in the reads tooltip that appears whenever the mouse hovers over a read. This provides at-a-glance details such as the read's name, its length, orientation, pair status (for paired-end data), and so on.
3. Tablet supports various methods of “packing” read data that affects the resultant visualization. Use the *Read Packing* button on the *Home|Visual* section of the toolbar to toggle between packed (all reads compressed together), stacked (each read displayed on a unique line), or paired-end alternatives where the mate pairs are visually linked before packing occurs.
4. The control panel contains various tabs to access functionality such as selecting the visible contig, selecting and highlighting features or annotations, providing information on all reads currently on screen, providing information on read groups, and finally searching for reads across the current contig or the entire data set.
5. The canvas tracker is displayed above the main canvas and presents elements such as the reference sequence (if provided), features/annotations, a scale-bar, position summaries, and protein translations. These views are optional and can be disabled to allow a larger main canvas view.
6. Above the canvas tracker is the overview area. This provides a scaled-to-fit visual summary of all of the (currently loaded) contig's data, either in the form of a read layout similar to the main display or in the form of a coverage histogram that shows read depth across the contig. Further options for it can be found in the *Advanced|Overview* section of the toolbar (see **Note 3**).
7. The BAM bar only appears if a BAM file is in use, whereupon it provides a graphical method to navigate around larger contigs when only small subsections are loaded into Tablet's memory at once (see **Note 4**).

8. Finally, the menu can be used to access features such as configuring advanced options for how Tablet runs. Look for this option under *Preferences* in the app menu if using OS X though.

2.4 Example Data

To aid the reader, the data for this chapter's use cases is available for automatic downloading within Tablet itself.

1. Click on the *OpenAssembly* button located in the *Home|Data* section of the toolbar.
2. Next, select the hyperlink labeled: "Unsure how to get started? Click here to open an example assembly".
3. Tablet will now present you with a list of example assemblies, among which will be five entries associated with the title of this book.
4. Select the appropriate entry for the use case in question and select *Open* to download it into Tablet.

3 Methods

3.1 Visual Validation of Variants in NGS Data

Visual validation of spot samples of variants from NGS data is an important application of assembly visualization software. This should form part of post-variant discovery quality control and will give the user an indication of the overall quality of the data generated. Primary variant discovery, however, should always be carried out using dedicated variant discovery software due to both the scale of the data and the complexity of the process (*see Note 5*). False positive/negative variants can be identified using a number of features within Tablet, and this can be used to fine-tune parameters used for variant calling and filtering.

1. Start by loading your assembly data into Tablet as described earlier.
2. If a GFF3 file with variant positions is available, import this into Tablet too. You can navigate through the list of variants in each contig by selecting it from the features list in the control panel. This will move the main canvas to the variant and highlight it for a few seconds.
3. Move the *Variants* slider on the *Home|Adjust* section of the toolbar to the right to highlight bases which differ from the reference sequence. This should appear to give the effect of darkening bases which match the reference, whilst highlighting bases that differ from the reference sequence. These will include sequencing errors and genuine variants.
4. You can also select the *Variants* color scheme either by clicking the *Read Colours* button on the *Home|Visual* section of the

toolbar and selecting it from the pop-up menu, or by directly selecting the scheme from the *Colour Schemes|Read Colouring* section of toolbar. This scheme colors bases matching the reference grey and those differing from the reference red. As you scroll around the display, any column that has a substantial proportion of its bases highlighted in red is a potential candidate SNP. This allows the user to obtain an initial assessment of the density and distribution of variants in a given dataset (*see* Fig. 2a).

5. The *Direction* color scheme displays forward reads in light green, whereas reads in the reverse orientation are colored blue (*see* Fig. 2b). This scheme is useful in examining individual SNPs for symptoms of systematic sequencing error in Illumina data. This type of systematic error is associated with both inverted repeats and with certain motifs in the sequence [6, 7]. Systematic error manifests itself as locations with extreme strand bias among the reads containing the alternate allele. In Tablet's default nucleotide color scheme these locations will be inconspicuous and appear to be normal, robust SNPs. With the *Direction* read color scheme, reads are colored by forward and reverse strand direction, and in cases of systematic error it becomes apparent that most, if not all, of the reads containing the alternate allele came from the same strand (*see* Fig. 2b). Variant calling software does not necessarily filter for this, and consequently additional annotation of variants for this is necessary with tools such as *syscall* [7], with additional visual spot checks in Tablet recommended for validation.

3.2 Assessing Reference Sequence Contiguity in Paired Read Mappings

SAM/BAM mappings of paired reads are typically annotated by the read mapper with respect to whether a given read is properly paired, or whether its mate is unmapped, mapped in a different contig, or at the wrong distance or orientation. Tablet extracts this information from SAM/BAM files and can use the *Read Type* color scheme to present the paired reads colored according to their classification. The default color scheme for this is as follows:

1. Properly paired reads: green (forward/R1 read) and blue (reverse/R2 read)
2. Improperly paired reads (mate in different contig): yellow (forward/R1 read) and pink (reverse/R2 read)
3. Orphaned reads (mate unmapped): red

In this example (*see* Fig. 3a), it is apparent that improperly paired reads (yellow and pink) are prevalent at the start and end of this contig. This suggests that the contig is a fragment of a whole chromosome, and that the chromosome has been assembled incompletely. Reads mapped to either end are paired with reads in

Sub Figures are Overlapped, Please Check

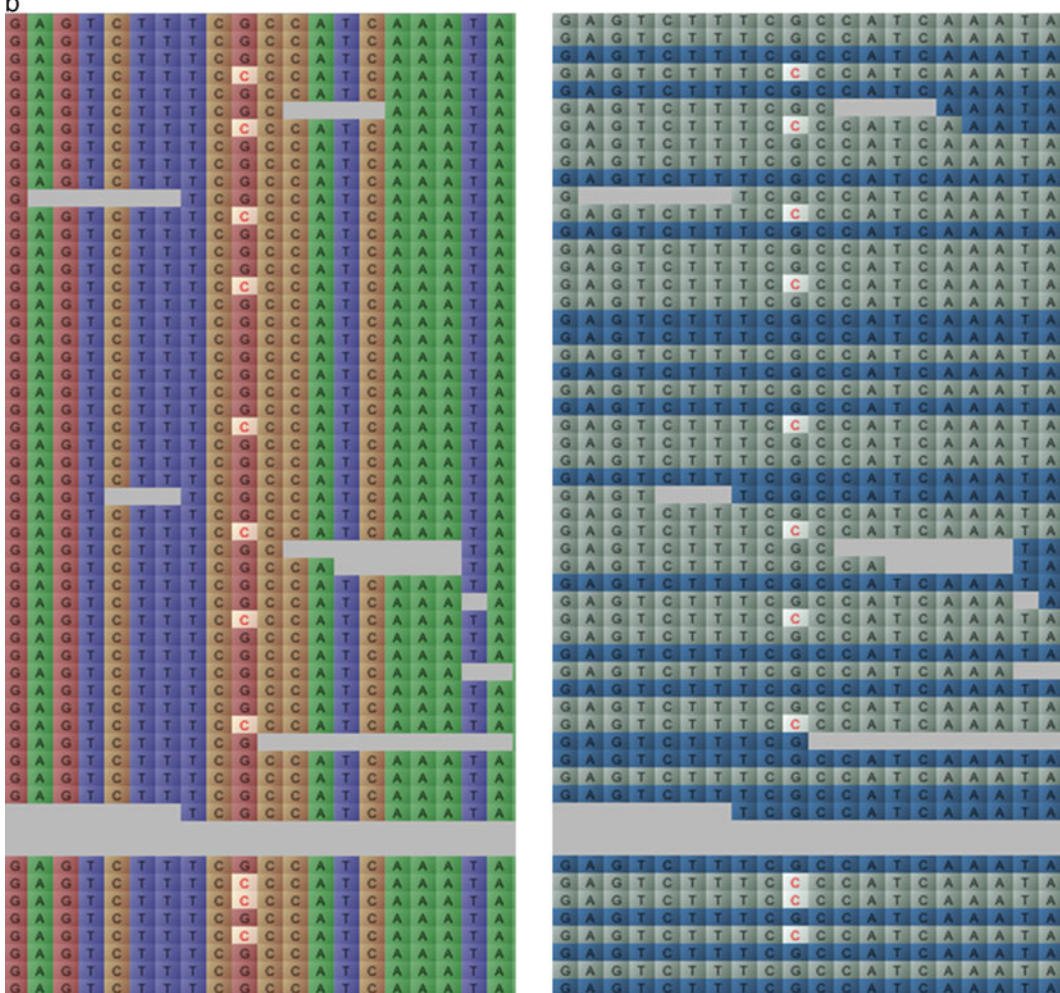
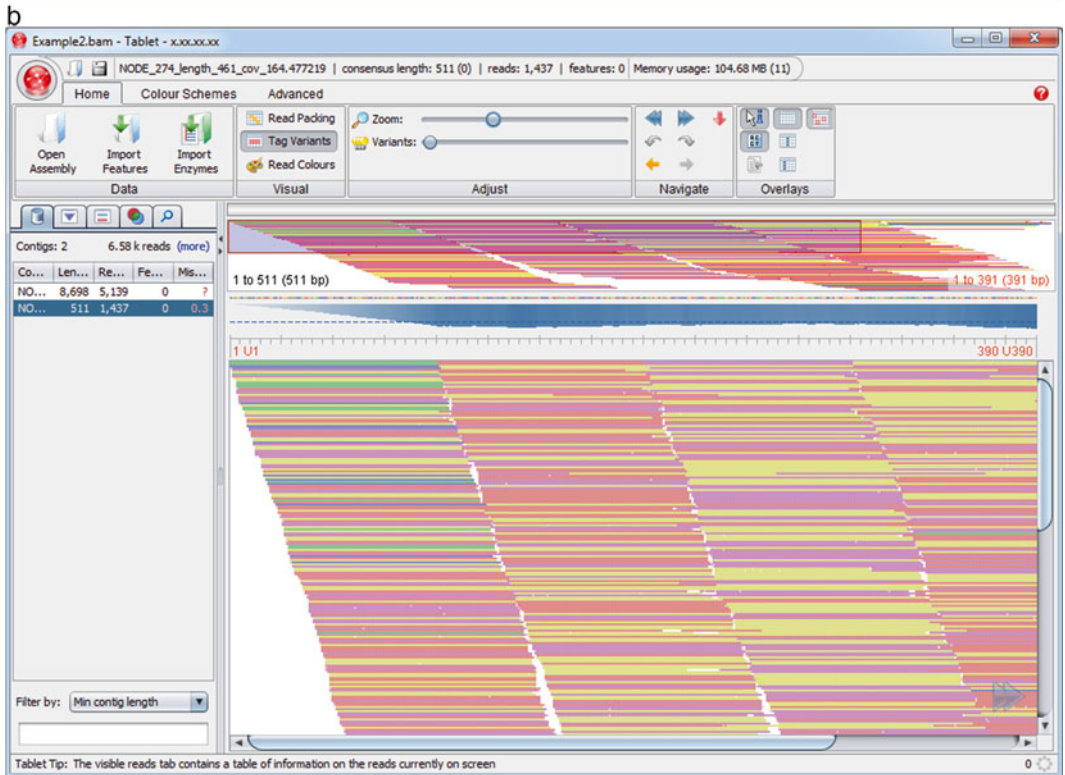
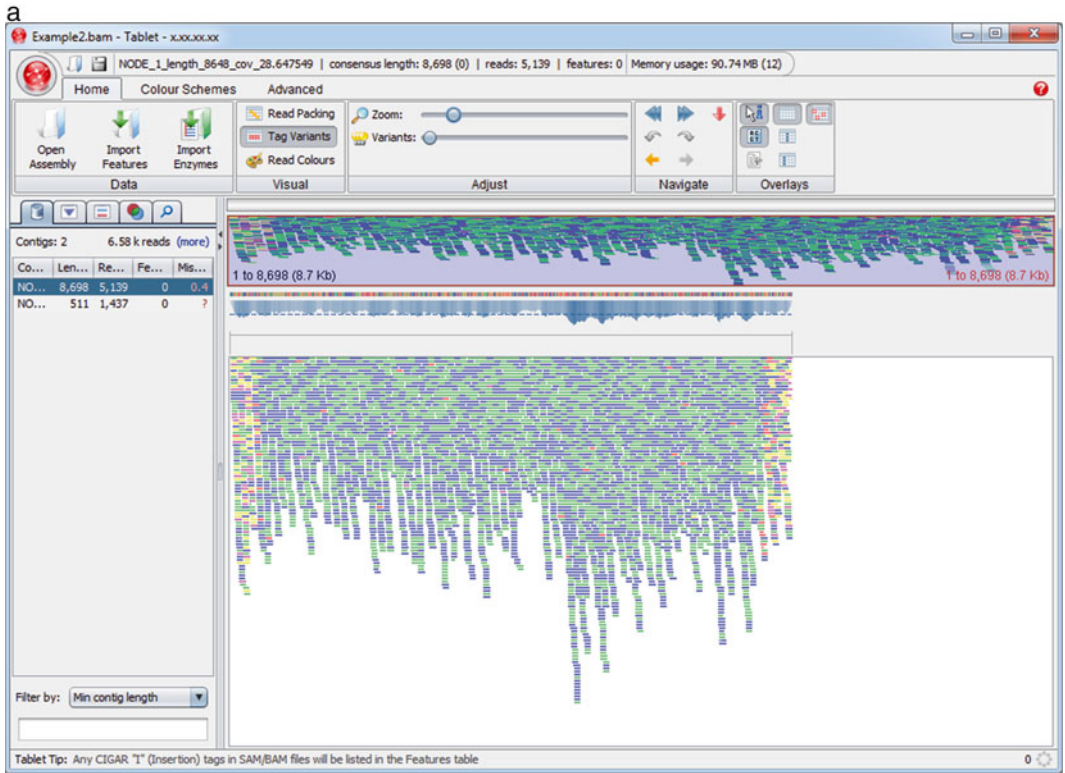


Fig. 2 (a) Illumina short read mapping, colored up with the *Variants* color scheme. Variant bases that differ from the reference sequence are highlighted in *red*. Two SNPs (Single Nucleotide Polymorphisms) are clearly visible as *red dotted vertical lines*. The remainder of the bases colored red represents read errors. Read data from [4], reference sequence from [5]. (b) Illumina systematic sequencing error, as revealed by the *Direction* color scheme. The figure on the *left* shows a SNP in the default *Nucleotide* color scheme. The figure on the *right* shows the same SNP but with the *Direction* color scheme selected. In this color scheme, forward reads are colored in *green*, while reverse reads are shown in *blue*. It is clear from the *right figure* that there is extreme strand bias, with all of the alternate alleles located on forward orientation reads only. This is one of several diagnostic features of Illumina systematic sequencing error [6, 7]. Dataset as in Fig. 2a

other contigs that represent the respective continuation of this contig, but have not been assembled together due to the presence of sequence features that prevent successful assembly (e.g., repeat sequences, polymorphisms, low complexity regions, gene families, pseudogenes).



Assuming a suitable mapping file contained paired-end data has been loaded and a contig selected, then the steps needed to reproduce this are as follows:

1. Select the Read Type color scheme from the *Colour Schemes|Read Colouring* section of toolbar.
2. Using the zoom slider on the *Home|Adjust* section of the toolbar, zoom out until you can see the entire contig on the main canvas. This will be evident by the absence of scroll bars and also from the reference sequence which should then be visible in its entirety. If this is not possible because the contig is too long, navigate to the start and the end separately by using the scrollbars, and if necessary, the BAM window slider above the overview canvas.
3. You can then inspect the read colors which should provide an indication of whether or not most reads are properly or improperly paired. If improperly paired reads are prevalent, you can use the mouseover tooltip to inspect these: the bottom half of the tooltip will display “Mate is not in this contig”, and will give details of which contig the mate is actually located in. When inspecting a number of the improperly paired reads, this information should consistently point to the same contig. This can be valuable information for cases where, e.g., a gene of interest is split over several contigs, and this evidence can be used to at least gather together its incompletely assembled component fragments.

As a further example of this (*see* Fig. 3b) here we have a contig where the contig length is considerably less than the average fragment length of the paired-end fragments, and this has resulted in a mapping where the vast majority of reads are improperly paired. Only a handful of reads are colored green/blue, and these represent reads from the lower end of the fragment size spectrum that have indeed been paired properly.

Fig. 3 (a) Paired-end read mapping, colored up with the *Read Type* color scheme. This color scheme renders properly paired reads in *green* (forward) and *blue* (reverse), while improperly paired reads are rendered in *yellow* (forward) and *pink* (reverse). Reads whose mate has not been mapped at all are shown in *red*. The reference sequence in this example represents a *de novo* assembled genomic contig from an assembly of data from the *E. coli* reference strain. The prevalence of reads colored *yellow*, *pink*, or *red* at the ends of the contig reflects the fragmentation of the assembly, as read pairs spanning the ends of the contig are not properly paired with their respective mates which are located in other contigs or unmapped. Data from NCBI Short read archive (<http://www.ncbi.nlm.nih.gov/sra/?term=ERR022075>). **(b)** A short contig taken from the same dataset. The vast majority of reads in this contig are colored *yellow/pink*, indicating improper pairing. This is a consequence of the contig being shorter than the majority of fragments in the sequencing library, which leads to paired reads getting mapped in different contigs. Contigs of this type are typical of highly fragmented genomic assemblies. Data as for part figure (a)

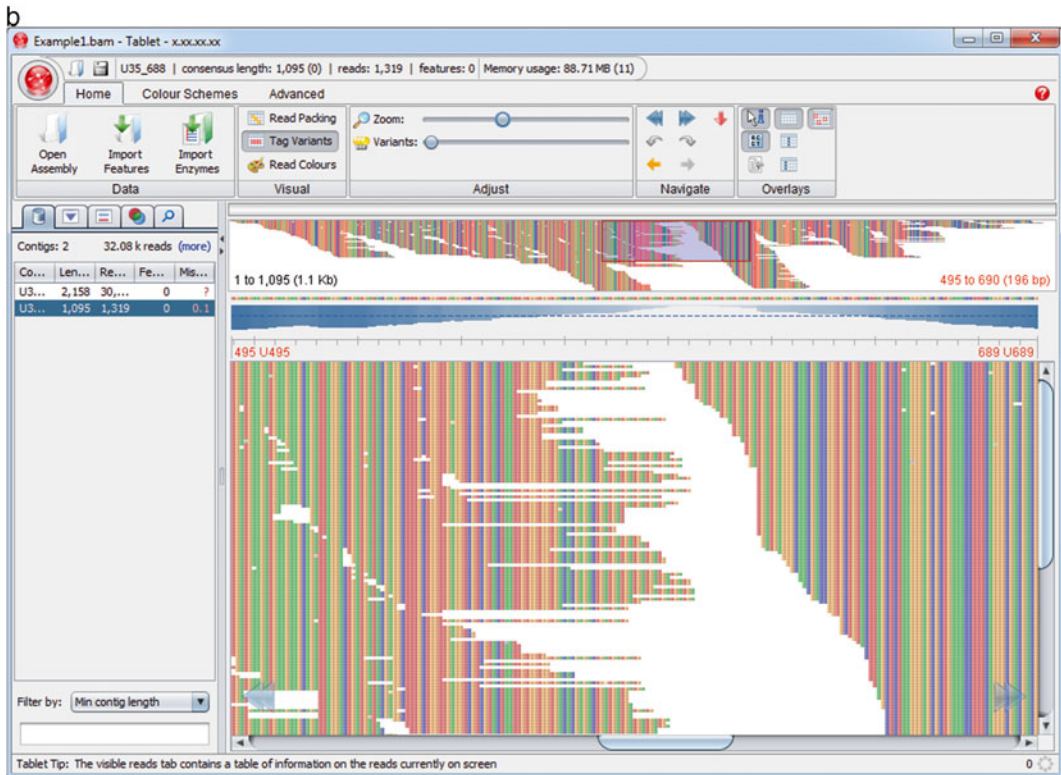
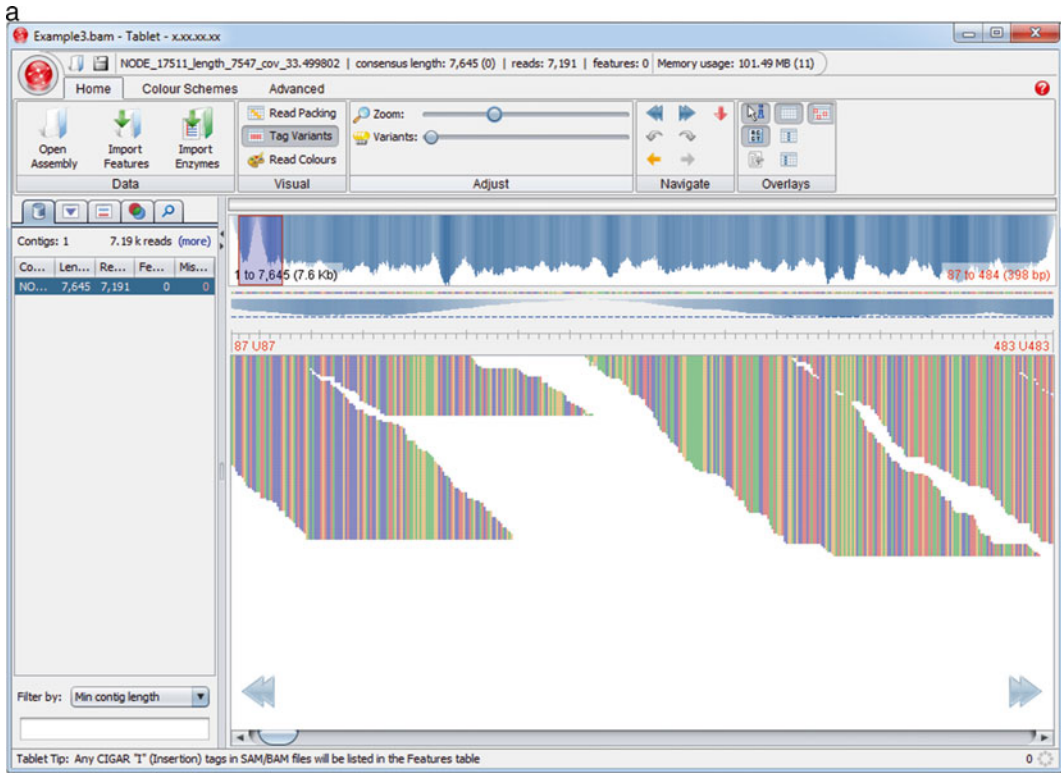
3.3 Visual Confirmation of Misassembly Using Coverage Dips

De novo assembly of NGS reads is complex and problematic, and frequently results in misassemblies. Misassemblies can be visualized by mapping the reads used for the assembly onto the assembled contigs, the assumption being that correct assembly will result in a relatively even distribution of the reads across contigs. Misassembled sequences disrupt the mapping of reads and this is visually apparent by sudden drops in read coverage because reads cannot be mapped across the misassembled stretch of reference sequence. In this example (*see* Fig. 4a) we show an example of misassembly as evidenced by a drop of read coverage from approx. 100× to several reads only, with no single read actually spanning the misassembled region fully. As another example (*see* Fig. 4b) we have a similar case, but this time the misassembly is related to the presence of a microsatellite sequence. These regions are notoriously difficult to assemble for graph-based short read assemblers as complexity is low and regions identical to this one may be found many times throughout the entire genome. The nucleotide color scheme is useful to flag this up as it renders the microsatellite as a repeating pattern that clearly stands out against the flanking sequence.

To detect low coverage regions,

1. Open the BAM file containing the read mapping as described above in the “Loading data” section.
2. The color scheme chosen for viewing the data is relatively unimportant in this case, as coverage levels are evident in all of these.
3. The overview canvas can display either scaled reads (the default) or a coverage graph. To change to the latter, right-click anywhere in the overview canvas and select *Coverage overview* from the context menu.
4. Visually scan the overview for obvious coverage dips. You can move the main canvas view onto these quickly by single-clicking on them in the overview canvas.

Fig. 4 (a) Coverage drops in read alignments such as the one shown here often indicate misassembly of the reference sequence. The data shown here is an alignment of short reads onto a de novo assembled reference sequence made from the same reads. None of the reads in this region fully spans the low coverage section, which is contrary to expectation as the reference sequence and at least some of the reads should match each other exactly throughout the entire assembly. Data from the *A. thaliana* TAIR resource (ftp://ftp.arabidopsis.org/home/tair/Sequences/whole_chromosomes/), reads simulated with the Sherman read simulator (<http://www.bioinformatics.babraham.ac.uk/projects/sherman/>). **(b)** Another coverage dip representing reference sequence misassembly. In this case the misassembly is the result of the presence of a microsatellite region which is evident as a repeating pattern of *green*, *red*, and *orange* bases either side of the low coverage region. Microsatellite sequences are notoriously difficult to resolve for de novo assemblers due to the low complexity of the sequence, and their presence frequently results in misassembly. Data from [4]



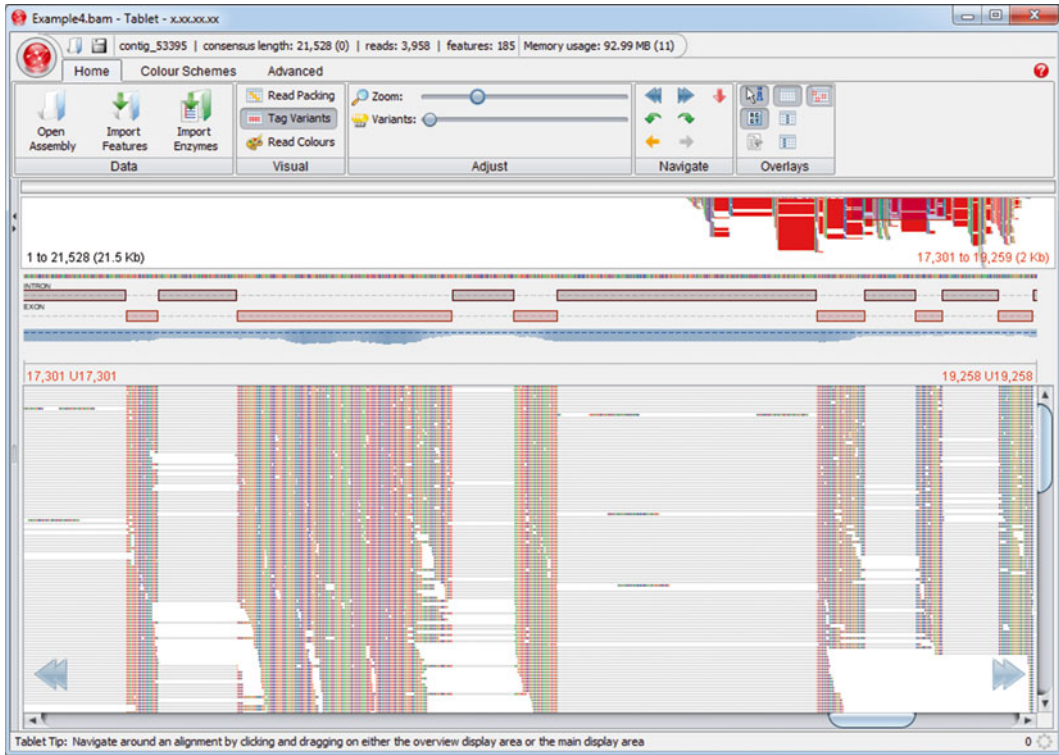


Fig. 5 A splice mapping of Illumina RNASeq reads mapped onto a genomic reference sequence. Reads were aligned to the reference sequence using the Tophat splice mapper [9], and the Augustus ab initio gene finder [10] was then used to generate gene prediction annotations. These were imported into Tablet as GFF files and are visible as feature tracks above the main canvas. The read mapping itself visualizes intronic gaps in the reads as *light grey* stretches of “N” characters that connect the exonic parts of reads. This visualization of the exon-intron structure is supported by the exon-intron annotation generated by Augustus, as evident by the agreement of the exonic parts of the reads with the EXON annotation track (*shown in pink*), and the agreement between the intronic “N” gaps which align with the INTRON track (*shown in dark purple* above the EXON track). Data from [11]

3.4 Visualization of Intron-Exon Structure with Splice Mappings and Annotation Tracks

One of the major advances involving next-generation sequencing is RNAseq, the high-throughput sequencing of RNA samples [8]. RNASeq has specific requirements for read mapping as it is generated from spliced transcripts which don’t contain introns. Mapping of RNASeq reads to a genomic reference sequence therefore requires the use of splice mapping software which produces split mappings for those reads in a sample that overlap splice junctions. Splice mappers such as Tophat [9] produce BAM files in which the resulting gaps in reads are filled with “N” characters, and these are readily visualized in Tablet as lighter stretches of characters (*see Fig. 5*).

Splice mappings can be supplemented with annotation tracks generated on the reference sequence by ab initio gene prediction programs such as Augustus [10]. In most cases annotation tracks are supplied in the generic feature format (GFF3) (*see* <http://www.sequenceontology.org/resources/gff3.html>), and these can

be imported directly into Tablet. Visualization of these tracks is extremely valuable for the purpose of mutual corroboration of the RNAseq evidence and the gene predictions.

This involves the following steps:

1. A BAM produced by a splice mapper is required (a list of splice mappers is available at http://seqanswers.com/wiki/Special:BrowseData/Bioinformatics_application?Bioinformatics_method=Mapping&Biological_domain=RNA-Seq_Alignment). Load this data, its reference sequence, and the GFF3 file as described earlier in the chapter.
2. In the control panel, select the *Features* tab then click on the *Select tracks* link. A dialog opens where you can select which annotation tracks you would like to visualize by ticking the appropriate checkboxes. Make sure the *intron* and *exon* tracks are selected, then click *OK*.
3. Above the main canvas the selected annotation tracks should now be visible. Barring intron retention and alternate splice patterns, the intron features should coincide with stretches of lightly colored N characters in split reads, and exon features should be aligned with read sequence.

3.5 Visualizing Read Provenance in Multi-sample Mappings

A variety of different applications can require that multiple samples are combined in a single BAM file together, rather than individual BAM files. For example, variant calling in NGS data requires that the variant calling tool has information on all the reads from all the samples for a candidate variant location. Support for this is provided in Tablet through a color scheme that utilizes SAM/BAM format read groups.

Read groups are a feature of the SAM/BAM specification that allow individual reads to be tagged with respect to a number of variables, such as sample name or sequencing platform (*see* <http://samtools.sourceforge.net/SAM1.pdf> for the SAM/BAM format specification). Read group tags are added to BAM records in the optional field, and this can either be done by the mapping tool itself, or post-mapping by tools such as *samtools merge* (<http://samtools.sourceforge.net/>). If read groups have been added, Tablet can assign different colors to each of these and these can then be visualized with the Read Group color scheme.

The required steps are as follows:

1. Ensure that your BAM file contains multiple samples and an appropriate header with read group entries. The reads in the BAM file also need to contain read group (RG) tags. If this is not the case, follow the instructions provided later (*see* **Note 6**).
2. From the *Read Colours* button on the *Home|Visual* tab on the toolbar, or the *Colour Schemes* tab in the toolbar, select the *Read Group* option. Reads will now be colored according to their read group.

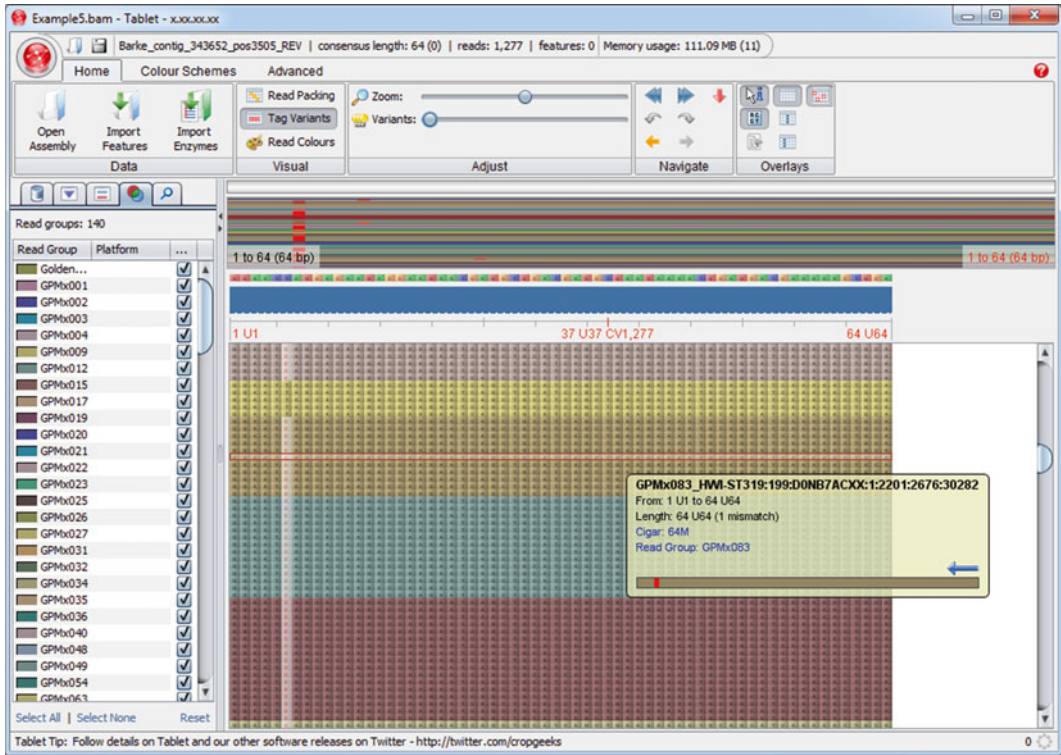


Fig. 6 Read alignment of genotyping-by-sequencing (GBS) reads to a reference sequence generated from a short stretch of genomic DNA adjacent to a restriction enzyme cut site. The color scheme used is the *Read Group* color scheme and reflects, in this case, the identity of the sample a read was derived from. The lightly colored bases represent alternate alleles. A single nucleotide polymorphism is clearly evident near the start of the alignment, and several samples are clearly homozygous for the alternate allele at the SNP site. Data from [12]

3. In the control panel, click on the *Read Groups* tab. You will see a list of all read groups available in the BAM file, along with the color they are presented by. You can customize the colors by double-clicking on the colored rectangle of a read group which brings up a color chooser window.
4. You can also click *Select None* at the bottom of the *Read Groups* tab—all the reads will be greyed out, and you can then selectively click the checkbox for those samples that you would like to color in. This can be useful for highlighting reads from individual samples.
5. Read group identifiers are also displayed in the tooltip when the cursor is moved onto a read.

Our final example (see Fig. 6) shows reads colored by read group, with each read group representing a different sample. This data was generated using genotyping-by-sequencing technology, where the reference sequences represent short stretches of genomic

DNA around restriction sites. Reference sequences and reads have been standardized to the same length, and this allows the reads from each sample to be grouped together which aids visualization greatly (read sorting is preserved when using the “samtools merge” utility). The lightly colored bases represent variants that differ from the reference sequence at this SNP location, and it is clear from the visualization that samples are homozygous for either the reference allele or the alternate allele.

4 Notes

1. If your operating system supports it, you may be able to load data into Tablet by simply dragging and dropping any assembly, reference, and annotation files directly from a file browser into Tablet’s main interface.
2. It is worth noting that the sequence IDs in the GFF3 file should exactly match the contig names in the reference and assembly file, otherwise the annotation will not be loaded.
3. Quickly toggle between the overview displays by right-clicking on it. Many of Tablet’s UI components support additional options only accessible via a right-click, especially the main canvas.
4. Navigation when a BAM file is loaded is subtly different, as only the data from the currently loaded *Bam Window* is displayed, rather than all data within a contig. This allows for very rapid loading of subsections of data at the expense of losing the bigger picture. The grey area of the BAM bar shows both how much data is currently loaded from the entire contig (represented in blue) and also where that data is within the contig.
5. Variant discovery is a large and complex subject, and cannot be covered exhaustively in this chapter. We recommend as a primer the literature reviews provided in [13–15], all of which provide guidelines for the general approach to be used, as well as lists of variant calling software.
6. Assuming samples have been mapped individually, and no read groups have been added by the mapping tool, use the “samtools merge” utility to merge the individual BAM files (*see* <http://samtools.sourceforge.net/samtools.shtml> for detailed instructions). This involves preparing a header file which will be added to the merged BAM file. The header file needs to contain the following information for each sample (a single line for each sample, elements separated by tab characters):

- (a) The read group tag: “@RG”.
- (b) A unique identifier for the read group, e.g., “ID:sample1”. This needs to match the name of the BAM file for this sample as the read group tag will be inferred by samtools based on that (in this example, the BAM file would be named sample1.bam).
- (c) A sample name, e.g., “SM:sample1”.

Assuming that all BAM files that are to be merged are located in the current working directory, and assuming that our header file is called “header.txt”, the following command will merge all the BAM files and add the appropriate read group to each read, based on its original BAM file name:

```
samtools merge -frh header.txt myMergedFile.bam *.bam
```

If the merged BAM files were unsorted, sort and then index the merged BAM file (*see* <http://samtools.sourceforge.net/samtools.shtml> for detailed instructions), and then load it and its reference sequence as described previously.

References

1. Milne I, Stephen G, Bayer M et al (2013) Using tablet for visual exploration of second-generation sequencing data. *Brief Bioinform* 14(2):193–202
2. Milne I, Bayer M, Cardle L et al (2010) Tablet—next generation sequence assembly visualization. *Bioinformatics* 26(3):401–402
3. Li H, Handsaker B, Wysoker A et al (2009) The sequence alignment/map (SAM) format and SAMtools. *Bioinformatics* 25:2078–2079
4. Comadran J, Kilian B, Russell J et al (2012) Natural variation in a homolog of *Antirrhinum CENTRORADIALIS* contributed to spring growth habit and environmental adaptation in cultivated barley. *Nat Genet* 44:1388–1392
5. Close TJ, Bhat RR, Lonardi S (2009) Development and implementation of high-throughput SNP genotyping in barley. *BMC Genomics* 10:582
6. Nakamura K, Oshima T, Morimoto T et al (2011) Sequence-specific error profile of Illumina sequencers. *Nucleic Acids Res.* doi:10.1093/nar/gkr344
7. Meacham F, Boffelli D, Dhahbi J et al (2011) Identification and correction of systematic error in high-throughput sequence data. *BMC Bioinformatics* 12:451
8. Wang Z, Gerstein M, Snyder M (2009) RNA-Seq: a revolutionary tool for transcriptomics. *Nat Rev Genet* 10:57–63
9. Trapnell C, Pachter L, Salzberg SL (2009) TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics*. doi:10.1093/bioinformatics/btp120
10. Stanke M, Waack S (2003) Gene prediction with a Hidden-Markov Model and a new intron submodel. *Bioinformatics* 19(2): 215–225
11. The International Barley Genome Sequencing Consortium (2012) A physical, genetic and functional sequence assembly of the barley genome. *Nature* 491:711–716
12. Liu H, Bayer M, Druka A et al (2014) An evaluation of genotyping by sequencing (GBS) to map the *Breviaristatum-e* (*ari-e*) locus in cultivated barley. *BMC Genomics* 15:104
13. Kumar S, Banks TW, Cloutier S (2012) SNP discovery through next-generation sequencing and Its applications. *Int J Plant Genomics*. doi:10.1155/2012/831460
14. Nielsen R, Paul JS, Albrechtsen A et al (2011) Genotype and SNP calling from next-generation sequencing data. *Nat Rev Genet* 12:443–451
15. Pabinger S, Dander A, Fischer M et al (2014) A survey of tools for variant analysis of next-generation genome sequencing data. *Brief Bioinform* 15(2):256–278

Chapter 15

Analysis of Genotyping-by-Sequencing (GBS) Data

Sateesh Kagale, Chushin Koh, Wayne E. Clarke, Venkatesh Bollina, Isobel A.P. Parkin, and Andrew G. Sharpe

Abstract

The development of genotyping-by-sequencing (GBS) to rapidly detect nucleotide variation at the whole genome level, in many individuals simultaneously, has provided a transformative genetic profiling technique. GBS can be carried out in species with or without reference genome sequences yields huge amounts of potentially informative data. One limitation with the approach is the paucity of tools to transform the raw data into a format that can be easily interrogated at the genetic level. In this chapter we describe bioinformatics tools developed to address this shortfall together with experimental design considerations to fully leverage the power of GBS for genetic analysis.

Key words Genotyping, Genotyping-by-sequencing, GBS, RAD-seq, Next generation sequencing, Geneticvariation, Single nucleotide polymorphism, InDels, Reduced representation sequencing, Trimmomatic, Bowtie, SAMtools, GATK, Demultiplexing, Read mapping, UnifiedGenotyper, HaplotypeCaller, Minor allele frequency, Imputation, Haplotype

1 Introduction

It was a significant achievement when the first plant genome sequence of *Arabidopsis thaliana* was published in 2000 [1] and heralded the application of genomics tools to plant research. The choice of this first species, with one of the smallest plant genomes and limited dispersed repetitive DNA, was partly driven by the cost and efficiency of available sequencing technologies. Today the transformative advances in sequencing platforms and chemistries, which have led to dramatic reductions in cost per base, have played a major role in deciphering multiple complex genomes. To date as many as 55 plant genomes have been sequenced and made publicly available [2] (<http://www.phytozome.net/>). Combined with such reference genome sequences next generation sequencing (NGS) has allowed a multitude of new approaches to be applied to the identification, analyses, and visualization of fundamental genetic variation. Identifying and utilizing natural and induced

genetic variation remains a prime objective in plant research with important implications in population genetics, evolution, and crop breeding. The most abundant and perhaps most informative variation that can be exploited are single nucleotide polymorphisms (SNPs) that have proven ideal markers for the study of plant genomes [3].

A number of approaches have been described to capture genome wide natural and induced genetic variation by NGS. The majority of these approaches rely on the use of reduced representation, which delimits the portion of large and complex genomes to be assessed to a manageable size. Initially proposed by Altshuler et al. [4] reduced representation allowed a high density SNP map to be generated for a genome previously thought to be too large for such analyses. However, it has been the combination of reduced representation, NGS and multiple indexing of samples that has provided the ability to study extremely large genomes at reasonable cost. The relative simplicity and cost-effectiveness of the genotype-by-sequencing (GBS) approach has encouraged its application in multiple species, including both model and non-model plants [5–8]. Also the increased marker density that is offered has led to its growing use in the anchoring of genome sequence assemblies, effectively removing the necessity to generate expensive and error prone physical maps [9–11]. The only current limitation is the bioinformatic and computational burden that is generated, with regard to both data processing and storage.

GBS now takes many forms, the first GBS data was generated using restriction site associated DNA sequencing (RAD-seq) [12] which utilized a single restriction enzyme combined with shearing of the digested DNA to capture a suitable portion of the genome. By optimizing enzyme choice and eliminating the necessity for DNA shearing the Cornell group simplified the approach and allowed more extensive multiplexing, which reduced costs further [13]. There have been several modifications to the basic protocols, predominantly incorporating the use of two enzyme digestion, including 2b-RAD [14], ddRAD-seq [15], and a variant to the Cornell GBS approach by Poland et al. [6] that utilizes methylation sensitive enzymes to further reduce the representation of the target genome. There have been several reviews describing the different approaches to GBS in plants [16–19].

The common feature of all the approaches is the type and volume of data that is produced, since all have exploited the Illumina sequencing platforms, generating millions of sequence reads usually of 100 bp or less for each indexed sample. Thus the bioinformatics pipeline described in the following chapter would be applicable to any of the published protocols in either single-end or paired-end read format. All the methods can be used in the absence of a reference genome; however, the use of a reference genome is generally far more effective in ensuring the robust identification of genome wide SNPs. The following chapter will focus on the

analyses of GBS data where there is access to a complete or draft genome; although tools (*see* **Publicly Available Software and Tools for GBS**) that have been developed to analyze GBS in the absence of a reference genome are listed.

2 Materials

In this chapter, we discuss a Bioinformatics pipeline (Fig. 1) that is designed to identify genetic variants such as SNPs and insertions/deletions (InDels) from NGS data generated by most major RAD and GBS approaches. This pipeline uses a suite of publicly available software and custom Perl scripts. There are alternative pipelines that have been developed and are listed in **Publicly Available Software and Tools for GBS**.

2.1 Publicly Available Software and Tools for GBS

1. *Trimmomatic* (<http://www.usadellab.org/cms/?page=trimmomatic>) is a multithreaded command line tool that can be used for trimming adapter sequences and low quality regions from Illumina sequencing reads [20].
2. *Bowtie2* (<http://bowtie-bio.sourceforge.net/bowtie2/index.shtml>) is an ultrafast short read alignment tool that can be used for aligning sequencing reads against a reference genome [21]. It should be noted that other alignment tools are available for this application, most commonly BWA [22].
3. *SAMtools* (<http://samtools.sourceforge.net/>) is a package of utilities designed for manipulating alignments in the SAM (Sequence alignment/Map) or BAM (Binary alignment/Map) format, including sorting, merging, indexing, and generating alignments in a per-position format [23].
4. *BCFtools* (<http://samtools.github.io/bcftools/>) is a set of utilities that manipulate variant calls in the Variant Call Format (VCF) and its binary counterpart (BCF).
5. *GATK* (*GenomeAnalysis Toolkit*) *genotyper* (<http://www.broadinstitute.org/gatk/>) provides a wide variety of tools for variant discovery and genotyping [24–26].
6. *STACKS* (<http://creskolab.uoregon.edu/stacks/>) allows de novo assembly of short read GBS data and the identification of genetic variation in the absence of a reference genome [27].
7. *TASSEL-GBS* (<http://www.maizegenetics.net/>) is an implementation of a GBS analysis pipeline in the TASSEL software package [28].

2.2 In House Tools

A set of utility Perl scripts (listed in Table 1) were written to perform various tasks associated with data processing, read alignment, and SNPdiscovery. These scripts are open source and freely available upon request.

Table 1

List of utility Perl scripts designed to perform various tasks associated with genetic variant discovery using RAD-Seq and GBS data sets

Perl script	Utility
<i>util_barcode_splitter.pl</i>	Demultiplexes paired-end RADseq or GBS reads based on a perfect match to barcodes
<i>util_find_uniq_reads.pl</i>	Compares read sequences and removes duplicate reads
<i>bowtie2_extract_best_global_hit.pl</i>	Goes through the SAM files and identifies the best hit from multi-mapped reads as having the top most hit with at least $X=6$ (or a user defined cutoff) penalty score better than the runner up
<i>bowtie2_extract_best_local_hit.pl</i>	Goes through the SAM files and identifies the best hit from multi-mapped reads as having the top most hit with at least $X=12$ (or a user defined cutoff) penalty score better than the runner up
<i>filter_vcf.pl</i>	Performs filtering based on missing genotype and minor allele frequency

3 Methods

The basic workflow for variant discovery using NGS data generated by RAD-seq and GBS approaches can be divided into three sequential steps: (1) raw data processing, (2) read alignment to a reference genome or de novo assembly of the sequence tags, and (3) variant discovery and annotation. In general, these three steps are shared by most of the currently available genotyping pipelines. In the following subsections, each of these steps are reviewed to provide background information for the available bioinformatics tools that are customized to perform various tasks associated with these steps.

3.1 Raw Data Processing

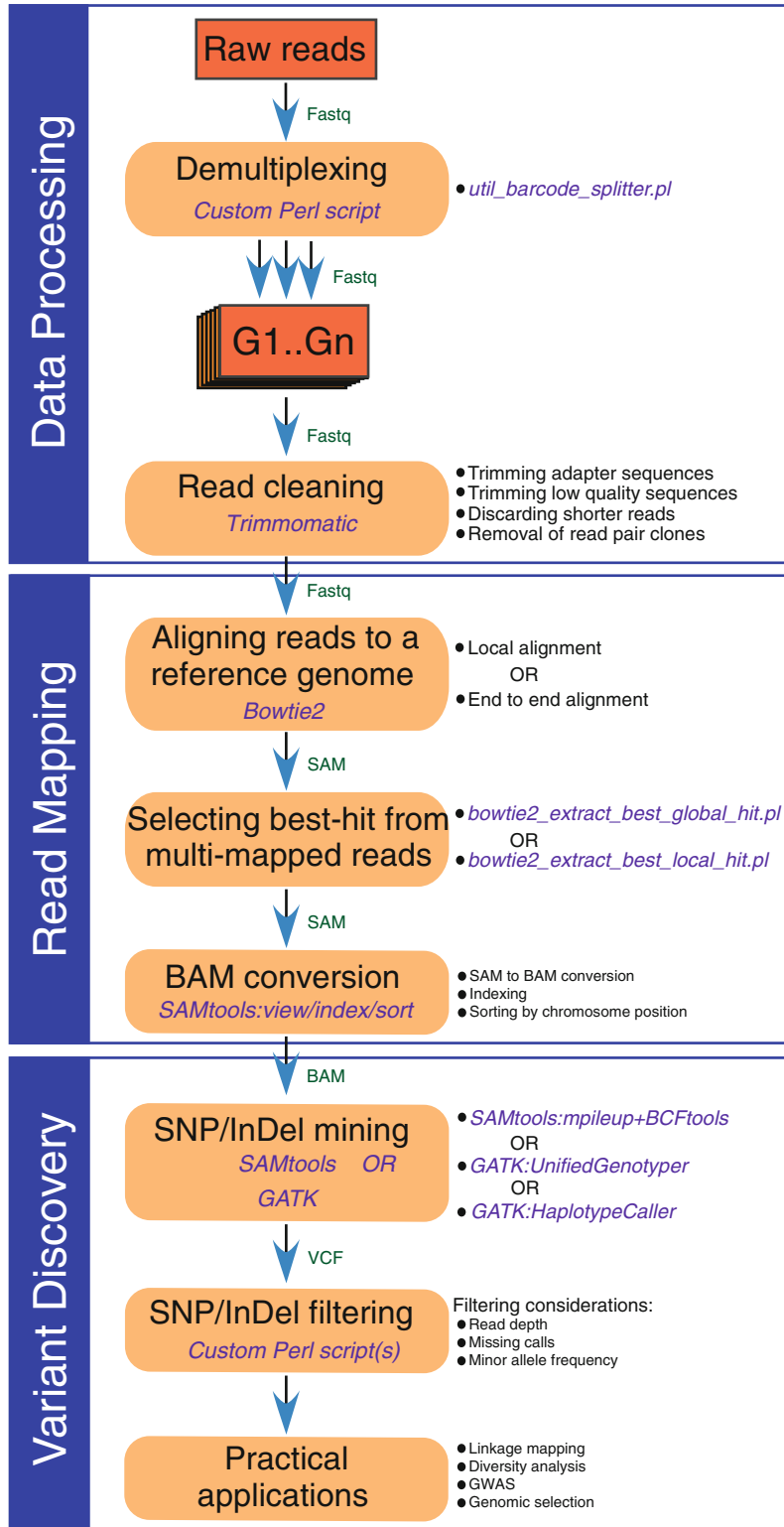
RAD-seq and GBS employ a highly multiplexed sequencing strategy for constructing reduced representation libraries for the Illumina NGS platform (*see Note 1*). Demultiplexing is the first key step of processing raw sequencing data, which separates reads into their corresponding samples based on barcode matching. Demultiplexing of Illumina reads is generally carried out using Illumina CASAVA or MiSeq reporter software; however, CASAVA cannot demultiplex RAD-seq and GBS reads which contain customized inline barcodes in only one of the adapter sequences. We have developed a Perl script *util_barcode_splitter.pl* (Table 1) to demultiplex RAD-seq and GBS reads.

Raw sequencing data often contain various types of errors and artifacts, such as base calling errors, low quality bases, adaptor contamination and duplicate reads [29]. Thus it is necessary to perform quality assessment and correction of reads by filtering or trimming of low quality reads or regions. There are numerous publicly available software that can be used for pre-processing of sequencing reads, such as Trimmomatic (<http://www.usadellab.org/cms/?page=trimmomatic>), PRINSEQ (<http://prinseq.sourceforge.net/>), FastqMcf (<http://code.google.com/p/ea-utils/wiki/FastqMcf>), FASTX-Toolkit (http://hannonlab.cshl.edu/fastx_toolkit/), and cutadapt (<http://code.google.com/p/cutadapt/>). In our pipeline (Fig. 1), we have adopted Trimmomatic, which is a fast, multithreaded command line tool that can be used to (1) remove adapter sequences, (2) trim leading and trailing low quality regions (below a user defined quality threshold), (3) scan the read with a user defined base-pair size sliding window and cut when the average quality per base has dropped below a threshold, and (4) keeping only those read-pairs where both reads were longer than the specified minimal length. Trimmomatic is also designed to handle “read-through” for paired-end data. A “read-through” is when a fragment size smaller than the read length is sequenced and hence results in overlapping read-pairs that include both the target fragment and adapter sequence. It is essential to remove one of the reads in this case in order to avoid over-stating read-depth for variant calling.

Amplification by polymerase chain reaction (PCR) is often used for target enrichment during the preparation of libraries for next-generation sequencing. PCR duplicates resulting from the original DNA templates being sequenced many times can have a detrimental effect on the quality of variant calls especially when the coverage is low (*see Note 2*). Computational methods for the detection and removal of PCR duplicates have become available that generally rely on the observation of identical alignment positions of reads to the reference genome. Read mapping being a computationally intensive process (*see Note 3*), the development of an alternate method for detection of PCR duplicates based on direct comparison of read sequences is essential, especially when the proportion of PCR duplicates is very high. To this end, we have developed a Perl script *util_find_uniq_reads.pl* (Table 1) that compares read sequences and removes duplicate reads.

3.2 Read Alignment to a Reference Genome

After read cleanup, alignment of short reads to a reference genome is the first step in a high-throughput genotyping workflow. In the absence of a reference genome, paired-end sequencing data generated by RAD-seq or GBS approaches can be assembled de novo using software packages such as STACKS [27], UNEAK [30], or RApiD [31] to produce mini-contigs that can be used as a reference for read mapping and genotyping (*see Note 4*). In the last few



years, a myriad of efficient short-read alignment programs, such as MAQ [32], mrsFast [33], STAMPY [34] Bowtie2 [35], BWA [22], and SOAP2 [36], have been developed. Most of these widely used aligners utilize hashing algorithms (MAQ, mrsFast, STAMPY) or Burrows–Wheeler transform (BWT) [37] based indexing (Bowtie2, BWA, and SOAP2) for short read mapping. The hash-based aligners use hash tables to store the information of either the reference genome or short reads. A major drawback of the hash-based aligners is that they require prohibitive amount of memory (*see Note 3*). The second generation BWT-based aligners are preferred as they consume only a limited amount of memory [38, 39].

In our genotyping workflow (Fig. 1), we have adopted Bowtie2 which is faster, more sensitive, and more accurate than BWA and SOAP2 across a wide range of parameter settings [35]. Bowtie2 supports both local and global (end-to-end) modes of alignment of short reads [35]. A local alignment considers only a short segment of the read and clips unaligned characters from one or both ends of the read to maximize the alignment score. Conversely, global alignment involves alignment of all characters in the read. In our experience, local mode of alignment of the reads is faster and useful for mapping reads generated by GBS, although less accurate (due to increased multi-mapping) than global alignment. GBS does not involve size fractionation of the sequencing library and hence sometimes results in the generation of fragments that are either too short to be useful or result in paired-end sequencing reads that overlap completely. On the other hand, the RAD-seq protocol includes a size fractionation step and most reads generated by this nonoverlapping approach can be aligned in an end-to-end manner. An example of the variation in the distribution of predicted enzyme sites for both RAD-seq (*EcoRI*) and GBS (*PstI* and *MspI*), together with a representation of relative genome coverage of each method, has been demonstrated for the *Brassica oleracea* genome [11]. RAD captured a greater portion of the genome with a high percentage of the potential sites being tagged and sequenced, while GBS coverage was impacted by the degree of cytosine methylation.

Multi-mapped reads are those that align to multiple locations within the reference genome sequence [40]. Most eukaryotic

Fig. 1 Bioinformatics workflow for genetic variant discovery using next generation sequencing based genotyping approaches such as RADseq and GBS. The genetic variant calling pipeline comprises three major steps, including raw data processing, read mapping to a reference genome, and variant discovery. Each of these steps is further divided into multiple sub-steps. The bioinformatics tools (shown in *purple*), input and output file formats (*green*), and the purpose, methodology, or general outcome of each sub-step (*bullet points*) in the workflow are presented

organisms, especially plants with polyploid genomes, carry orthologous and paralogous gene families that contain multiple isoforms of the same gene with nearly identical or similar sequences. Shorter reads being less specific tend to have more multi-mapping events. In polyploid plant species, the proportion of multi-mapped reads ranges from 20 to 60 %. Discarding such a high proportion of multi-mapping reads will result in a significant loss of valuable information. Bowtie2 searches and reports all valid alignments that score better than a given cutoff. We use Perl utility scripts *bowtie2_extract_best_global_hit.pl* or *bowtie2_extract_best_local_hit.pl* to go through the SAM files and identify the best hit from multi-mapped reads as having the top most hit with at least $X=6$ (end-to-end) or $X=12$ (local) penalty score better than the runner up. The larger the X score, the more confident a read is uniquely mapped but more alignments get discarded as a consequence.

Bowtie2 outputs alignments in SAM format which contains alignment data in human readable tab-delimited text. SAM files generally tend to be very large. BAM, a compressed binary version of SAM format, is a preferred format for the downstream variant detection analyses due to its relatively smaller size. We use the “*view*” command of SAMtools to convert mapped reads from SAM to BAM format. For downstream analysis the alignments in BAM files must be sorted and indexed according to the chromosomal positions. To achieve this, we use the sort and index utilities of SAMtools.

3.3 Variant Discovery

The next step after mapping reads to a reference genome is to call sequence variants (SNPs and InDels) from the processed BAM file. Multiple software tools for variant-calling are available, including SAMtools:mpileup/BCFtools [23], GATK [24–26], SOAP [41], SNVer [42], and GNUMAP [43]. A recent study performed systematic evaluation of these commonly used variant-calling bioinformatics pipelines and found a very poor concordance between variants called by each of these methods [44]. Each of the SNP calling methods is designed based on different sets of assumptions about the reference genome and reads, and their suitability in different situations depends upon various factors, including the nature of genotypes, presence or absence of multi-allelic SNPs, and sensitivity and specificity of detecting SNPs. In our variant-calling workflow, we have implemented two of the most commonly used SNP callers; SAMtools:mpileup/BCFtools [23] and GATK [24, 25]. Both of these pipelines also call InDels.

SAMtools:mpileup computes the likelihood of each possible genotype by generating a consensus sequence using the MAQ (Mapping and Assembly with Quality) model framework, which uses a general Bayesian framework for picking the base that maximizes the posterior probability with the highest Phred quality score, and outputs the information in the BCF format (binary

variant call format). However, it does not call the variants. BCFtools does the actual calling and estimating allele frequency by applying the genotype likelihood information in BCF files. It generates output in the VCF (variant call format) format, which is the emerging standard for storing variant data. Identification of InDels from paired-end reads is relatively more challenging than that of SNPs as incorrect placement of insertions or deletions during read alignment to a reference genome may lead to false positive SNPs. SAMtools:mpileup deploys a concept called Base Alignment Quality (BAQ; [45]) to provide an efficient and effective way to rule out false positive SNPs caused by alignment artifacts. With the BAQ strategy which is invoked by default in mpileup, the probability of a base being misaligned can be accurately measured. Although the combination of SAMtools:mpileup and BCFtools offers a straightforward way of calling SNPs and InDels, this approach is limited to only diploid calling as SAMtools:mpileup is designed to compute and handle only biallelic variants [45]. We have successfully used SAMtools:mpileup for variant-calling and genetic linkage mapping of populations produced from biparental crosses (Bollina et al., In preparation; [10, 11]).

GATK is similar to SAMtools but utilizes additional processing steps, such as local realignment around InDel loci in order to clean up alignment artifacts, marking non-informative duplicate reads, and quality recalibration of both base quality and variant quality to improve overall accuracy of variant-calling [24–26, 44]. GATK includes two variant calling tools, UnifiedGenotyper and HaplotypeCaller. The UnifiedGenotyper uses a Bayesian genotype likelihood model to estimate posterior probability of allele frequency at each locus. Additionally it utilizes information from multiple samples and supports SNP calling from non-diploid samples. The HaplotypeCaller, which combines a local de novo assembler with a more advanced hidden Markov model (HMM) likelihood function, outperforms the UnifiedGenotyper in discovering sequence variants. However, it currently supports only diploid calling and lacks multithreading support.

Filtering raw SNP candidates is an essential step in the genotyping workflow as it helps in reducing false positive calls made from biases in the sequencing data and removing those calls that do not fulfil specific thresholds for SNP and genotype properties. Filtering of false positive calls based on read depth and quality threshold is embedded within some of the currently available variant calling pipelines such as SAMtools and GATK. We perform additional filtering based on missing genotyping calls and minor allele frequency (MAF). The level of missing data depends upon sequencing coverage which is influenced by the multiplexing level and the output from sequencing platform [18, 46]. Missing data can be reduced by sequencing at higher depth and reducing the multiplexing level. An alternative method for replacing missing

data is to impute missing values with plausible substitutes (*see Note 5*). In recent years, algorithms [47–49] have been developed for imputation of missing genotype data with great accuracy. MAF refers to the frequency at which the least common allele occurs in a given population [50]. We use the Perl utility script *filter_vcf.pl* (Table 1) to perform filtering based on missing genotype and MAF generally ignoring SNPs with a MAF less than 5 %. The final output from the majority of the variant calling pipelines is generally in the VCF format which can be viewed using genomic viewers such as Tablet [51] or IGV [52] (Fig. 2). We have also developed Perl scripts to generate genotype scores in tab delimited file formats for ease of downstream processing and analysis. The last step of our genotyping workflow involves merging SNPs based on identical segregation patterns. The cartoon in Fig. 3 depicts the logic as well as our approach for creating haplotypes blocks by merging closely linked SNP markers with identical segregation patterns to provide a recombination bin framework that can be easily incorporated into genetic mapping analysis.

3.4 Conclusion

The advent of very high throughput NGS platforms together with new technical methodologies to take advantage of these gains provided an opportunity for establishing high resolution genetic analysis in any species. The ability to profile large numbers of targeted loci for sequence variation in highly multiplexed sets of discrete individuals provided a platform for a range of applications. An initial limitation for the full deployment of these approaches have been the dearth of readily available bioinformatics tools to process the raw data to yield output that can be readily incorporated into classical genetic analyses. This chapter has outlined some of the recently available bioinformatics resources to enable researchers to establish GBS applications for genetic analysis in their laboratories, provided an example pipeline that could be utilized for this purpose, and also a description of key factors that need to be considered in experimental design.

4 Notes

1. Assessing sequencing data requirements: In many instances both RAD and GBS have been attempted with a number of restriction enzymes. However, the choice of a particular enzyme and the volume of sequencing data required depends on several factors such as, the genome size, sample multiplexing needs, GC content, frequency of the cut site (frequent to rare) and desired frequency of the sites throughout the genome. In silico analysis of a genome with a choice of an enzyme cut site would provide a glimpse prior to a selection. The RAD Counter tool provided on the RAD wiki website (<https://www.wiki.ed.ac.uk/display/RADSequencing/Hom>

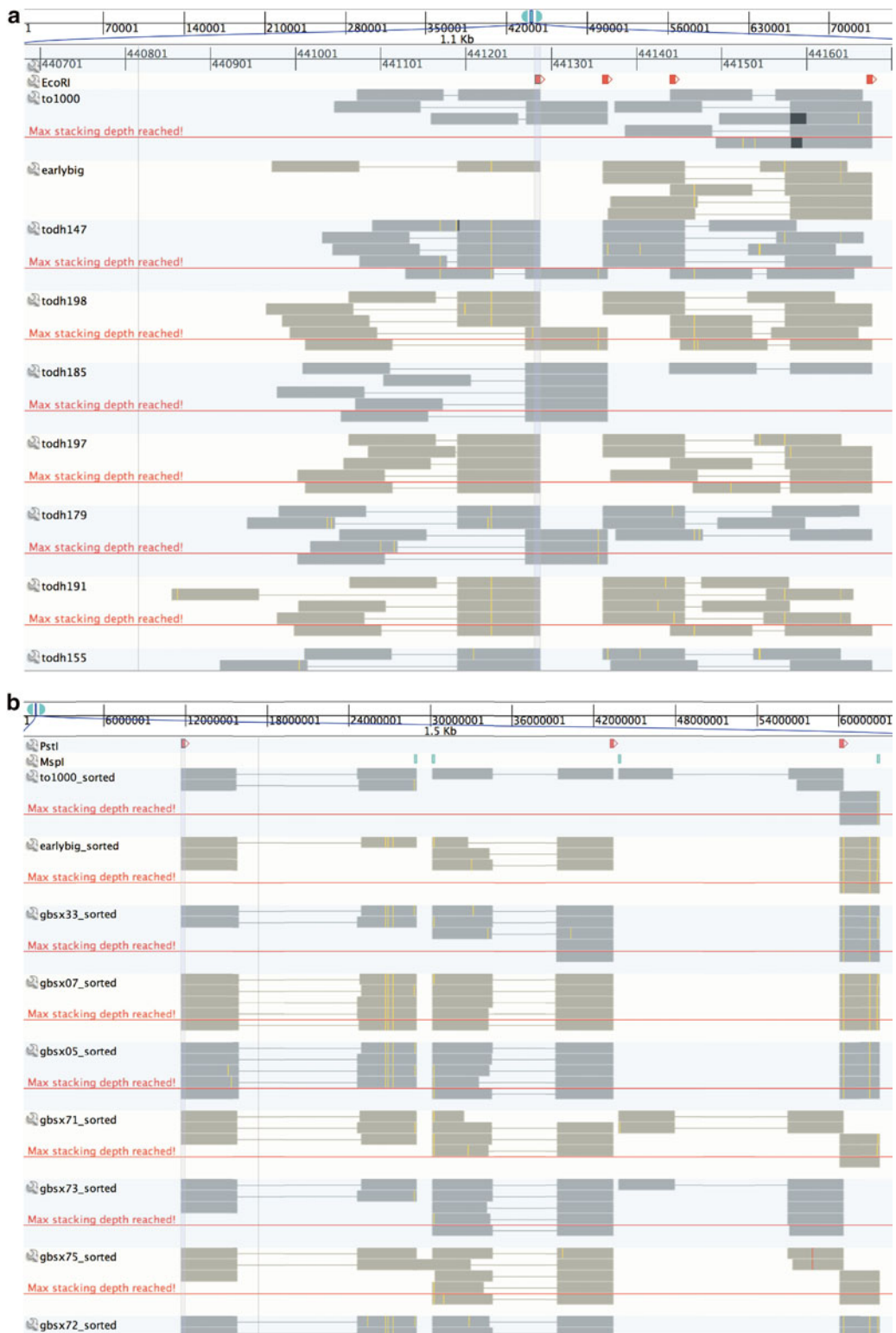


Fig. 2 Genomeviewer (IGV; Thorvaldsdottir et al. [52]) images illustrating alignment to the reference genome of short paired-end reads generated by RAD-seq (a) and GBS (b) approaches. The *top* two/three tracks represent the reference contig and positions of restriction site(s): *EcoRI* (RAD-seq) or *PstI* and *MspI* (GBS). The following tracks show reads from each individual library aligned back to the reference using Bowtie2. Read bases that match the reference are displayed in *gray* and those that do not match (sequence variants) are shown in *yellow*

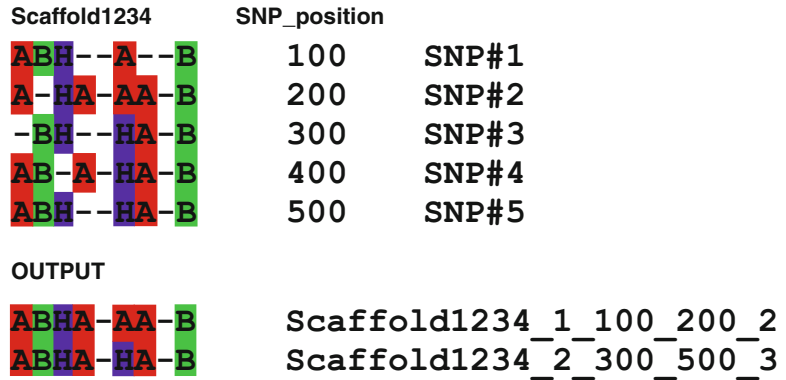


Fig. 3 Overview of the approach used for generating haplotypes by merging SNPs with identical segregation patterns. As per the example shown in this cartoon, 5 RAD SNPs (at positions 100, 200, 300, 400, and 500 bp) were identified on scaffold1234. SNP#1 and SNP#2 have identical segregation pattern, except for the missing data points, so as SNP#3 to SNP#5. Instead of using all 5 SNPs for genetic mapping, we combine SNPs with identical scores. The locus name of each merged RAD SNP (haplotype) provides additional information: the first part of the name includes the scaffold name, the next number indicates the order of the SNP pattern identified in the scaffold, the next two numbers indicate the base pair positions between which this haplotype pattern was found, and the final number indicates the count of independent SNPs that had this pattern

[ej;sessionid=14E3C4ECD753766FC8E4EA41274A9BF1](#)) provides the user with a simple Excel format to input relevant information with respect to the above parameters to establish the optimal experimental design to ensure appropriate read depth is reached.

2. Removal of duplicate reads: advantages and limitations: Duplicate reads arising from PCR amplification during library preparation can result in perfect copies of the DNA template being sequenced multiple times. The proportion of duplicate reads can vary enormously and duplicate reads can artificially inflate read coverage which may have detrimental effect on the quality of variant calls. Hence the dataset used for variant calling should include only one copy per duplicate set of reads. Duplicate reads can be detected and removed by comparison of either the read sequences or their alignment coordinates. However, the risk of removal of identical or almost identical reads arising from duplicated genomic regions, especially in organisms carrying polyploid genomes, poses a serious challenge. Additionally, it is impossible to differentiate duplicate reads arising due to amplification bias and identical GBS tags originating from the same restriction site(s) at a particular genomic location. This is not an issue in the case of paired-end

RAD tags as the additional DNA fragmentation combined with size fractionation step in RAD-sequencing protocol leads to the production of paired-end tags with at least one variable end. Thus we advise against removal of duplicate GBS tags, whereas the decision on removal of duplicate RAD tags should depend upon the ploidy status or the level of segmental duplication in the organism under consideration.

3. Computational resources: The analysis of GBS and RAD data requires nontrivial computational resources. In order to reduce analysis time, the use of multiple CPU cores is recommended. Many desktop computers will be limited in the number of samples they can process by the available RAM. Additionally, the output of the analysis steps requires significantly more hard disk space than that of the raw sequencing data. As an example of computational requirements, 96 GBS samples were processed using 16 CPU cores for Trimmomatic, Bowtie2, and GATK. The total time required to process the samples was approximately 13 h and required at most 21GB of RAM. The samples were demultiplexed from 9.7GB of compressed fastq data and resulted in approximately 68 GB of uncompressed output using a pipeline optimized to reduce production of intermediary output files.
4. Single-end or paired-end mapping: Variant calling can be done using either single or paired-end data with resulting benefits in increased coverage with paired-end data. It is also difficult to accurately map single reads originating from regions with significantly higher sequence homology, such as repeat rich or duplicated genomic regions. Sequencing reads from both ends can partly overcome this difficulty. Filtering of paired-end sequencing data based on adapter contamination and quality as well as length thresholds results in the generation of a small proportion of single end reads. In such case both single-end and paired-end mapping followed by merging of separately generated SAM files before the variant discovery step is possible.
5. Data imputation: One issue with both RAD and GBS is the amount of missing data that can result from the sequencing, especially when this is carried out at a low level of coverage/depth. Hopefully such an outcome can be avoided in the first place by ensuring optimal levels of depth are reached by adopting an appropriate experimental design (*see Note 1*). However, when high levels of missing data result it is possible to adopt imputation approaches that are currently available for different experimental approaches with various population structures [49, 53]. As well, it is possible to limit the amount of missing data in some types of populations; for example biparental genetic mapping populations as described in the main text.

In this case the merging of SNP loci based on identical segregation patterns can be carried out to create haplotypes blocks with minimal missing data and a resultant recombination bin framework for genetic mapping analysis.

References

1. The Arabidopsis Genome Initiative (2000) Analysis of the genome sequence of the flowering plant *Arabidopsis thaliana*. *Nature* 408(6814):796–815
2. Michael TP, Jackson S (2013) The first 50 plant genomes. *Plant Genome* 6:1–7
3. Ganai M, Altmann T, Roder M (2009) SNP identification in crop plants. *Curr Opin Plant Biol* 12:211–217
4. Altshuler D, Pollara VJ, Cowles CR, Van Etten WJ, Baldwin J, Linton L, Lander ES (2000) An SNP map of the human genome generated by reduced representation shotgun sequencing. *Nature* 407:513–516
5. Barchi L, Lanteri S, Portis E, Valè G, Volante A, Pulcini L, Ciriaci T, Acciarri N, Barbierato V, Toppino L, Rotino GL (2012) A RAD tag derived marker based eggplant linkage map and the location of qtls determining anthocyanin pigmentation. *PLoS One* 7, e43740
6. Poland JA, Rife TW (2012) Genotyping-by-Sequencing for plant breeding and genetics. *Plant Genome* 5:92–102
7. Wang N, Thomson M, Bodles WJA, Crawford RMM, Hunt HV, Featherstone AW, Pellicer J, Buggs RJA (2013) Genome sequence of dwarf birch (*Betula nana*) and cross-species RAD markers. *Mol Ecol* 22:3098–3111
8. Liu H, Bayer M, Druka A, Russell J, Hackett C, Poland J, Ramsay L, Hedley P, Waugh R (2014) An evaluation of genotyping by sequencing (GBS) to map the *Breviaristatum-e* (ari-e) locus in cultivated barley. *BMC Genomics* 15:104
9. Varshney RK, Song C, Saxena RK, Azam S, Yu S, Sharpe AG, Cannon S, Baek J, Rosen BD, Tar'an B, Millan T, Zhang X, Ramsay LD, Iwata A, Wang Y, Nelson W, Farmer AD, Gaur PM, Soderlund C, Penmetsa RV, Xu C, Bharti AK, He W, Winter P, Zhao S, Hane JK, Carrasquilla-Garcia N, Condie JA, Upadhyaya HD, Luo M-C, Thudi M, Gowda CLL, Singh NP, Lichtenzweig J, Gali KK, Rubio J, Nadarajan N, Dolezel J, Bansal KC, Xu X, Edwards D, Zhang G, Kahl G, Gil J, Singh KB, Datta SK, Jackson SA, Wang J, Cook DR (2013) Draft genome sequence of chickpea (*Cicer arietinum*) provides a resource for trait improvement. *Nat Biotechnol* 31:240–246
10. Kagale S, Chushin K, Nixon J, Bollina V, Clarke WE, Tuteja R, Spillane C, Robinson SJ, Links MG, Clarke C, Higgins EE, Huebert T, Sharpe AG, Parkin IAP (2014) The emerging biofuel crop *Camelina sativa* retains a highly undifferentiated hexaploid genome structure. *Nat Commun* 5:3706
11. Parkin IAP, Koh C, Tang H, Robinson SJ, Kagale S, Clarke WE, Town CD, Nixon J, Krishnakumar V, Bidwell SL, Denoeud F, Belcram H, Links MG, Just J, Clarke C, Bender T, Huebert T, Mason AS, Pires JC, Barker G, Moore J, Walley PG, Manoli S, Batley J, Edwards D, Nelson MN, Wang X, Paterson AH, King G, Bancroft I, Chalhoub B, Sharpe AG (2014) Transcriptome and methylome profiling reveals relics of genome dominance in the mesopolyploid *Brassica oleracea*. *Genome Biol* 15:R77
12. Baird NA, Etter PD, Atwood TS, Currey MC, Shiver AL, Lewis ZA, Selker EU, Cresko WA, Johnson EA (2008) Rapid SNP discovery and genetic mapping using sequenced RAD markers. *PLoS One* 3, e3376
13. Elshire RJ, Glaubitz JC, Sun Q, Poland JA, Kawamoto K, Buckler ES, Mitchell SE (2011) A robust, simple genotyping-by-sequencing (GBS) approach for high diversity species. *PLoS One* 6, e19379
14. Wang S, Meyer E, McKay JK, Matz MV (2012) 2b-RAD: a simple and flexible method for genome-wide genotyping. *Nat Methods* 9:808–810
15. Peterson BK, Weber JN, Kay EH, Fisher HS, Hoekstra HE (2012) Double digest RADseq: an inexpensive method for de novo SNP discovery and genotyping in model and non-model species. *PLoS One* 7, e37135
16. Davey J, Hohenlohe P, Etter P, Boone J, Catchen J, Blaxter M (2011) Genome-wide genetic marker discovery and genotyping using next-generation sequencing. *Nat Rev Genet* 12:499–510
17. Deschamps S, Llacá V, May GD (2012) Genotyping-by-Sequencing in plants. *Biology* 1:460–483
18. Poland JA, Brown PJ, Sorrells ME, Jannink JL (2012) Development of high-density genetic maps for barley and wheat using a novel two-

- enzyme genotyping-by-sequencing approach. *PLoS One* 7, e32253
19. Edwards D, Batley J, Snowdon R (2013) Accessing complex crop genomes with next-generation sequencing. *Theor Appl Genet* 126:1–11
 20. Bolger AM, Lohse M, Usadel B (2014) Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics*. doi:10.1093/bioinformatics/btu170
 21. Langmead B, Trapnell C, Pop M, Salzberg SL (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol* 10:R25
 22. Li H, Durbin R (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* 25:1754–1760
 23. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R, Genome Project Data Processing Subgroup (2009) The Sequence Alignment/Map format and SAMtools. *Bioinformatics* 25:2078–2079
 24. McKenna A, Hanna M, Banks E, Sivachenko A, Cibulskis K, Kernytsky A, Garimella K, Altshuler D, Gabriel S, Daly M, DePristo MA (2010) The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res* 20:1297–1303
 25. DePristo MA, Banks E, Poplin R, Garimella KV, Maguire JR, Hartl C, Philippakis AA, del Angel G, Rivas MA, Hanna M, McKenna A, Fennell TJ, Kernytsky AM, Sivachenko AY, Cibulskis K, Gabriel SB, Altshuler D, Daly MJ (2011) A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nat Genet* 43:491–498
 26. Van der Auwera GA, Carneiro MO, Hartl C, Poplin R, del Angel G, Levy-Moonshine A, Jordan T, Shakir K, Roazen D, Thibault J, Banks E, Garimella KV, Altshuler D, Gabriel S, DePristo MA (2013) From FastQ data to high-confidence variant calls: the genome analysis toolkit best practices pipeline. *Curr Protoc Bioinform* 43:11.10.1–11.10.33
 27. Catchen JM, Amores A, Hohenlohe P, Cresko W, Postlethwait JH (2011) Stacks: building and genotyping loci *de novo* from short-read sequences. *G3* 1:171–182
 28. Glaubitz JC, Casstevens TM, Lu F, Harriman J, Elshire RJ, Sun Q, Buckler ES (2014) TASSEL-GBS: a high capacity genotyping by sequencing analysis pipeline. *PLoS One* 9, e90346
 29. Dai M, Thompson RC, Maher C, Contreras-Galindo R, Kaplan MH, Markovitz DM, Omenn G, Meng F (2010) NGSQC: cross-platform quality analysis pipeline for deep sequencing data. *BMC Genomics* 11 Suppl 4: S7
 30. Lu F, Lipka AE, Glaubitz J, Elshire R, Cherney JH, Casler MD, Buckler ES, Costich DE (2013) Switchgrass genomic diversity, ploidy, and evolution: novel insights from a network-based SNP discovery protocol. *PLoS Genet* 9, e1003215
 31. Willing EM, Hoffmann M, Klein JD, Weigel D, Dreyer C (2011) Paired-end RAD-seq for *de novo* assembly and marker design without available reference. *Bioinformatics* 27:2187–2193
 32. Li H, Ruan J, Durbin R (2008) Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Res* 18:1851–1858
 33. Hach F, Hormozdiari F, Alkan C, Hormozdiari F, Birol I, Eichler EE, Sahinalp SC (2010) mrsFAST: a cache-oblivious algorithm for short-read mapping. *Nat Methods* 7:576–577
 34. Lunter G, Goodson M (2011) Stampy: a statistical algorithm for sensitive and fast mapping of Illumina sequence reads. *Genome Res* 21:936–939
 35. Langmead B, Salzberg SL (2012) Fast gapped-read alignment with Bowtie 2. *Nat Methods* 9:357–359
 36. Li R, Yu C, Li Y, Lam TW, Yiu SM, Kristiansen K, Wang J (2009) SOAP2: an improved ultrafast tool for short read alignment. *Bioinformatics* 25:1966–1967
 37. Burrows M, Wheeler DJ (1994) A block-sorting lossless data compression algorithm. Systems Research Center Research Report 124, Digital Systems Research Center, Palo Alto, CA.
 38. Li H, Homer N (2010) A survey of sequence alignment algorithms for next-generation sequencing. *Brief Bioinform* 11:473–483
 39. Huang L, Popic V, Batzoglou S (2013) Short read alignment with populations of genomes. *Bioinformatics* 29:i361–i370
 40. Fonseca NA, Rung J, Brazma A, Marioni JC (2012) Tools for mapping high-throughput sequencing data. *Bioinformatics* 28:3169–3177
 41. Li R, Li Y, Fang X, Yang H, Wang J, Kristiansen K, Wang J (2009) SNP detection for massively parallel whole-genome resequencing. *Genome Res* 19:1124–1132
 42. Wei Z, Wang W, Hu P, Lyon GJ, Hakonarson H (2011) SNVer: a statistical tool for variant calling in analysis of pooled or individual next-generation sequencing data. *Nucleic Acids Res* 39, e132

43. Clement NL, Snell Q, Clement MJ, Hollenhorst PC, Purwar J, Graves BJ, Cairns BR, Johnson WE (2010) The GNUMAP algorithm: unbiased probabilistic mapping of oligo-nucleotides from next-generation sequencing. *Bioinformatics* 26:38–45
44. O’Rawe J, Jiang T, Sun G, Wu Y, Wang W, Hu J, Bodily P, Tian L, Hakonarson H, Johnson WE, Wei Z, Wang K, Lyon GJ (2013) Low concordance of multiple variant-calling pipelines: practical implications for exome and genome sequencing. *Genome Med* 5:28
45. Li H (2011) Improving SNP discovery by base alignment quality. *Bioinformatics* 27:1157–1158
46. Andolfatto P, Davison D, Erezylmaz D, Hu TT, Mast J, Sunayama-Morita T, Stern DL (2011) Multiplexed shotgun genotyping for rapid and efficient genetic mapping. *Genome Res* 21:610–617
47. Marchini J, Howie B, Myers S, McVean G, Donnelly P (2007) A new multipoint method for genome-wide association studies by imputation of genotypes. *Nat Genet* 39:906–913
48. Browning BL, Browning SR (2009) A unified approach to genotype imputation and haplotype-phase inference for large data sets of trios and unrelated individuals. *Am J Hum Genet* 84:210–223
49. Huang BE, Raghavan C, Mauleon R, Broman KW, Leung H (2014) Efficient imputation of missing markers in low-coverage genotyping-by-sequencing data from multi-parental crosses. *Genetics* 197:401–404
50. Robinson MR, Wray NR, Visscher PM (2014) Explaining additional genetic variation in complex traits. *Trends Genet* 30:124–132
51. Milne I, Bayer M, Cardle L, Shaw P, Stephen G, Wright F, Marshall D (2010) Tablet—next generation sequence assembly visualization. *Bioinformatics* 26:401–402
52. Thorvaldsdottir H, Robinson JT, Mesirov JP (2013) Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. *Brief Bioinform* 14:178–192
53. Marchini J, Howie B (2010) Genotype imputation for genome-wide association studies. *Nat Rev Genet* 11:499–511

Skim-Based Genotyping by Sequencing Using a Double Haploid Population to Call SNPs, Infer Gene Conversions, and Improve Genome Assemblies

Philipp Emanuel Bayer

Abstract

Genotyping by sequencing (GBS) is an emerging technology to rapidly call an abundance of Single Nucleotide Polymorphisms (SNPs) using genome sequencing technology. Several different methodologies and approaches have recently been established, most of these relying on a specific preparation of data. Here we describe our GBS-pipeline, which uses high coverage reads from two parents and low coverage reads from their double haploid offspring to call SNPs on a large scale. The upside of this approach is the high resolution and scalability of the method.

Key words Genotyping by sequencing, SNPs, SNPcalling, Bioinformatics, Genotyping

1 Introduction

Genotyping by sequencing (GBS) is a new technology which employs the huge amounts of sequence reads generated by modern sequencing technology to genotype Single Nucleotide Polymorphisms (SNPs). GBS calls an abundance of SNPs, which is useful in several different applications: trait mapping, genome-wide association studies, the construction of genetic maps, description of recombination patterns, and more. Several different GBS protocols have been introduced and used in the last few years.

The earliest GBS-protocol is RAD sequencing (RAD-Seq). RAD sequencing uses restriction enzymes to over-sequence DNA close to restriction sites, allowing a reduced representation of genomes [1]. This protocol has been used to create RAD tags as a basis for microarrays carrying 2000 markers in three-spined stickleback. However, the technology carries several biases, notably a bias towards certain restriction sites, a bias away from calling SNPs at heterozygous restriction sites, and a small bias towards

over-sequencing RAD loci with higher GC content [2]. To counter the restriction enzyme bias, other protocols use two restriction enzymes with different restriction sites [3]. This approach mapped 34,000 SNPs in barley and 20,000 in wheat.

Other protocols directly use genome sequencing reads. The benefit of using genomic reads is that the aforementioned biases do not exist, and that almost all regions of the genome are sampled. However, there is no reduced representation of genomes, and other types of biases may be introduced. For example, Illumina reads are under-sampled from regions of low or high GC content [4], and PacBio RSI reads have a high error rate [5], possibly leading to a higher rate of false positive SNPs. One study used reads with an average coverage of 0.02× to call 1,493,461 SNPs for 150 recombinant inbred lines of *Oryza sativa* [6]. Another study used a much higher coverage (on average: 20.6× per plant) to call 425,357 SNPs in 40 *Arabidopsis thaliana* plants [7]. Another variant of GBS uses AFLP to reduce genomic complexity, and it was used to call 1409 SNPs in *A. thaliana* and 5583 SNPs in *Lactuca sativa* (lettuce) [8].

Here we introduce our implementation of GBS which uses whole-genome sequencing reads from two parental individuals and their double haploid offspring population.

2 Materials

2.1 Sequencing Reads

Two sets of sequencing reads are required. To call SNPs, high coverage reads from both parents are needed (*see* **Notes 1** and **2**). Ideally the coverage starts from 30× to 40×. To call genotypes this protocol uses low coverage reads from double haploid offspring (*see* **Note 3**). The coverage usually ranges from 1× to 7×. So far, this protocol has only been tested with Illumina GAI, Miseq, and Hiseq reads, but it should work with other sequencing technologies, as well. Reads need to be either in FASTQ or FASTA format.

2.2 Reference Genome

All reads need to be aligned to a reference genome. How well assembled the genome has to be is dependent on the needs of the researcher (*see* **Note 4**). It is possible to assemble contigs based on genomic reads, and then to align the reads against these contigs. This might, however, lead to sampling biases in the read dataset. The reference needs to be in fasta format.

2.3 Software

Our GBS workflow uses several different publicly available tools. To align reads against the reference, we use SOAPaligner [9] (*see* **Note 5**). Samtools [10] and Picard tools (<http://picard.sourceforge.net/>) are used to convert file-formats, clean alignments, and more. SGSautoSNP [11] is used to call SNPs from alignments (*see* **Note 6**). Two simple in-house scripts are used to compare SNPs to alignments. Finally, Flapjack [12] is used to visualize genotypes, and Tablet [13] is used to visualize alignments.

3 Methods

3.1 Alignment of Reads to Reference Using SOAPaligner

First, all reads of the two parents and of the double haploid population have to be aligned to the reference genome. If there are several different libraries, it makes sense to run one alignment per library, since the insert sizes often differ. We use SOAPaligner in this step, making sure that reads only aligning uniquely are retained.

```
soap -p 4 -r 0 -m 0 -x 1000 -a reads_R1.fastq -b reads_R2.fastq
-D reference.fasta.index -o paired_output.soap -2 unpaired_out-
put.soap -u unmapped_output.soap
```

This will run soap with four threads, an insert size from 0 to 1000 and will only retain uniquely aligning reads. Reads aligning without being paired have a higher chance of having been mapped in the wrong position, so only reads aligning in pairs are kept from this step. The following steps all require SOAPaligner's output to be in bam-format, so the output's format is changed using soap2sam.pl (included with SOAPaligner) and samtools.

```
soap2sam.pl paired_output.soap>paired_output.sam samtools
faidx reference.fasta samtools view -bt reference.fasta.fai paired_out-
put.sam>paired_output.bam samtools index paired_output.bam
```

This step will result in one bam-file per library.

3.2 Sorting and Merging Using Picard and Samtools

For later steps, it is important to keep track which reads came from which cultivar. In our case, we add a tag for each cultivar to the front of read-names—for example, we add *TPI-* to the front of each read name for the cultivar Tapidor. This can be done with sed:

```
sed -i.bak 's/^\s*/TPI-/' your_alignment.sam
```

This will create a backup of the original file ending in *.bak* and overwrite *your_alignment.sam*. Next, the libraries for parental individuals are merged and sorted by positions. In this example, parent A has three libraries, and parent B has only one library.

```
java -Xmx2g -jar /path/to/MergeSamFiles.jar
INPUT=parent_A_lib_1.bam INPUT=parent_A_lib_2.bam
INPUT=parent_A_lib_3.bam INPUT=parent_B_lib_1.bam
OUTPUT=parents_merged.bam SORT_ORDER=coordinate
CREATE_INDEX=true
```

At this point it is good practice to investigate the merged bam-file using samtools or Tablet to make sure that all reads have been renamed correctly, and that both cultivars are present. For the double haploid population, it is not necessary to rename the reads, or to merge all files. Only when there are several libraries present for one DH-individual is merging needed, but all files need to be sorted, for example for two individuals TN01 and TN02:

```
java Xmx2g -jar /path/to/MergeSamFiles.jar INPUT=TN01_
lib1.bam INPUT=TN01_lib2.bam OUTPUT=TN01_merged.
bam SORT_ORDER=coordinate CREATE_INDEX=true java
Xmx2g -jar /path/to/SortSam.jar INPUT=TN02_lib1.bam
OUTPUT=TN02_lib1_sorted.bam SORT_ORDER=coordinate
CREATE_INDEX=true
```

This step will result in one sorted bam-file per individual of the double haploid population, and one sorted bam-file containing alignments of both parents.

3.3 Removal of Clonal Reads and PCR Clones Using Picard

Clonal reads may unnecessarily raise SNP scores of sequencing errors, leading to false positive SNPs. Therefore, we remove all clonal reads from the parental alignment files using Picard tools.

```
java -Xmx2g -jar /path/to/MarkDuplicates.jar INPUT=your_alignment.bam OUTPUT=your_cleaned_alignment.bam REMOVE_DUPLICATES=true METRICS_FILE=metrics.txt
```

3.4 Calling SNPs Using SGSautoSNP

SGSautoSNP [11] is a reference-free SNPcaller (in other words: the nucleotide on the reference at the position of a possible SNP does not matter). The standard version of SGSautoSNP does not allow heterozygosity in parental individuals. In other words, only SNPs between the two parental individuals are reported; if one individual carries two or more alleles at the position of a SNP, the SNP is discarded.

SGSautoSNP is written in Python 2.7 and can be run in two ways. The first is to run it with the entire dataset and specify several threads—each thread will process one reference chromosome or contig. The second is to split up the bam files into chromosomes and to run one SGSautoSNP instance per file. Which way to choose is a matter of personal preference and given resources. Researchers with access to clusters and job arrays might prefer the second way. Here, the first way is presented. SGSautoSNP is run with the merged parental alignments, the reference the alignments were created with, as well as a *gff3* file detailing where contigs begin and end on the given reference chromosomes.

```
SGSautoSNP.py --bam parents_merged.bam --fasta reference.fasta --contig_output contig_snps --chr_offset contigs.gff3 --chr_output chromosome_output --cultivars "A,B" --cpu 1
```

This will create *gff3* and VCF files starting with “contig_snps” and “chromosome_output”, along with a few files detailing statistics like the total number of SNPs. It is of course possible to use other SNPcalling programs in this step.

3.5 Calling Genotypes Using snp_genotyping_all.pl

Our in-house script *snp_genotyping_all.pl* is run to call all genotypes for all individuals of the double haploid population. For each individual, the script compares the bases aligning at all known SNP positions. If the bases at that position are all identical to the allele of parent A, the genotype A is assigned. If all bases at the position are identical to the allele of parent B, the genotype B is assigned. When there are several different alleles, the genotype is heterozygous and an E is assigned. When no reads are present, no genotype is assigned (in our case, denoted by a -). Here is an example with an alignment of the individual TN01, SNPs in the file *Reference_snps.gff3*, known contig positions in *Reference_contig.gff3*, and the cultivars “A” and “B”.


```
snp_genotyping_all.pl -b TN01_merged.bam -gff Reference_
snps.gff3 -off Reference_contig.gff3 -f Reference.fasta -o
Reference_TN01 -c1 A -c2 B -h AB -s 1
```

This will generate three output files per individual, each detailing the distribution of genotypes in a different way.

3.6 Visualizing Genotypes

Flapjack is used to display the distribution of genotypes in the population. First, all results of `snp_genotyping_all.pl` have to be collated and transformed into the format Flapjack uses. This is done using our in-house script `parseGenotype.pl`, run once per reference chromosome/contig.

```
parseGenotype.pl -out reference reference/*_csv
```

This will parse positions and genotypes from the output of subsection 3.5 and collate the data in a *dat* and a *map* file, ready for import into Flapjack.

3.7 Cleaning SNPs

There is a good chance that the above procedure results in false positive SNPs. One way to remove false positive SNPs is to remove all SNPs that do not segregate in the population. This is done by iterating over all SNPs, counting the genotypes from all individuals, and removing all SNPs that show only one genotype. In our experience, this removes about 10 % of all SNPs (data not shown). Not all of the SNPs removed are false positives—in some cases, these SNPs are located in regions that are hard to sequence due to, for example, a high GC count. Since the parents are sequenced at a much higher rate, the likelihood to sequence these regions is much higher than in the double haploid population. Thus, only few genotypes are called in the double haploid population for this particular SNP, and the SNP looks like a false positive SNP, while it is in reality a false negative.

3.8 Imputing Genotypes

It is possible to impute missing genotypes for individuals. Since recombinations are assumed to be rare, an individual's missing genotype should correspond to the surrounding genotypes. Therefore, researchers can impute missing genotypes by inserting surrounding genotypes, and leaving them missing if the surrounding genotypes conflict. Imputation of genotypes by comparing an individual to other individuals, like it is done in MACH [14] or Impute [15], is possible too.

4 Notes

1. Parental individuals need to be as homozygous as possible. Heterozygous SNPs are removed from the analysis by this protocol, since it is difficult to assign a parental genotype to them. Therefore, this protocol is geared towards plants with little heterozygosity.

2. The relatedness of parental individuals matters. If both parents are closely related, the number of SNPs between them will be very low, too low to gather insights about the distribution of genotypes in the offspring population. A low genotype resolution means that in the offspring population, SNPs associated with phenotypes, as well as smaller recombination events will not be observed.
3. In some populations, not all individuals are actually double haploid. These individuals exhibit a much higher rate of heterozygous alleles than others, and exhibit a much higher number of recombinations. It is unclear why this happens since there are several possible explanations. It could be that some double haploid plants are contaminated with pollen from other plants during the selfing step. Contamination during any step of the data preparation and processing could be another source of this phenomenon. For the time being, we remove all individuals with many heterozygous alleles: the exact cut-off depends on the dataset, the reference organism, the population, the naturally occurring heterozygosity and much more.
4. The quality of the reference genome matters in some cases. For example, researchers using scaffolded contigs might run into problems when many of the contigs are misplaced, in which case paired reads are not aligned properly (read A aligns in completely different region than read B). Some tools exist to catch a glimpse of a reference genome's quality, e.g., CEGMA [16] or REAPR [17]. It is up to the researcher to find and fix problems in the reference sequence.
5. The choice of the alignment algorithm matters. Short read aligners differ in their sensitivity and accuracy [18], as well as in the way duplicate alignments are assessed and reported. The user has to make sure that with each tool, the number of reported reads aligning multiple times is as little as possible. SNPs based on reads that align in multiple regions may exist in reality, but their exact position is a choice out of several alternatives. Therefore, for most applications these SNPs should not be called in the first place. Different aligners handle the removal of multiply aligning reads differently. For example, SOAPaligner has the `-r 0` option which suppresses the output of reads aligning in several places. Users should be aware that some settings are treacherous—the `-g 1` option of TopHat [19] just reports one alignment if there are several alignments for a read, making it harder to filter out reads that align non-uniquely. With most read aligners, filtering the resulting alignments using samtools [10] is the easiest approach. In most alignment programs, reads mapping in several regions usually receive a low mapping quality, here is an example to use samtools to remove all reads with a mapping score below 20: `sam-`

```
toolsview -q 20 -b your_alignment.bam > your_filtered_alignment.
bam
```

Other programs like BWA [20] set the *XA* flag when a read aligns multiple times, so that all reads carrying this flag can be removed using *grep -v XA*. Researchers are advised to check the manual of their aligner of choice.

6. The choice of SNP calling algorithm matters. The number of called SNPs as well as sensitivity and specificity vary greatly between SNP calling programs [21, 22]. For researchers who want stringent results it may make sense to use only SNPs that are called by at least two different SNP calling programs.

Acknowledgement

The author acknowledges funding support from the Australian Research Council (Project LP1f10100200). Support is also acknowledged from the Queensland Cyber Infrastructure Foundation (QCIF) and the Australian Partnership for Advanced Computing (APAC).

References

1. Miller MR, Dunham JP, Amores A et al (2007) Rapid and cost-effective polymorphism identification and genotyping using restriction site associated DNA (RAD) markers. *Genome Res* 17:240–248. doi:10.1101/gr.5681207
2. Davey JW, Cezard T, Fuentes-Utrilla P et al (2013) Special features of RAD sequencing data: implications for genotyping. *Mol Ecol* 22:3151–3164. doi:10.1111/mec.12084
3. Poland JA, Brown PJ, Sorrells ME, Jannink J-L (2012) Development of high-density genetic maps for barley and wheat using a novel two-enzyme genotyping-by-sequencing approach. *PLoS One* 7:e32253. doi:10.1371/journal.pone.0032253
4. Chen Y-C, Liu T, Yu C-H et al (2013) Effects of GC bias in next-generation-sequencing data on de novo genome assembly. *PLoS One* 8:e62856. doi:10.1371/journal.pone.0062856
5. Carneiro MO, Russ C, Ross MG et al (2012) Pacific biosciences sequencing technology for genotyping and variation discovery in human data. *BMC Genomics* 13:375. doi:10.1186/1471-2164-13-375
6. Huang X, Feng Q, Qian Q et al (2009) High-throughput genotyping by whole-genome resequencing. *Genome Res* 19:1068–1076. doi:10.1101/gr.089516.108
7. Yang S, Yuan Y, Wang L et al (2012) Great majority of recombination events in Arabidopsis are gene conversion events. *Proc Natl Acad Sci U S A* 109:20992–20997. doi:10.1073/pnas.1211827110
8. Truong HT, Ramos AM, Yalcin F et al (2012) Sequence-based genotyping for marker discovery and co-dominant scoring in germplasm and populations. *PLoS One* 7:e37565. doi:10.1371/journal.pone.0037565
9. Li R, Yu C, Li Y et al (2009) SOAP2: an improved ultrafast tool for short read alignment. *Bioinformatics* 25:1966–1967. doi:10.1093/bioinformatics/btp336
10. Li H, Handsaker B, Wysoker A et al (2009) The Sequence Alignment/Map format and SAMtools. *Bioinformatics* 25:2078–2079. doi:10.1093/bioinformatics/btp352
11. Lorenc MT, Hayashi S, Stiller J et al (2012) Discovery of single nucleotide polymorphisms in complex genomes using SGSautoSNP. *Biology* 1:370–382. doi:10.3390/biology1020370
12. Milne I, Shaw P, Stephen G et al (2010) Flapjack—graphical genotype visualization. *Bioinformatics* 26:3133–3134. doi:10.1093/bioinformatics/btq580

13. Milne I, Bayer M, Cardle L et al (2010) Tablet—next generation sequence assembly visualization. *Bioinformatics* 26:401–402. doi:[10.1093/bioinformatics/btp666](https://doi.org/10.1093/bioinformatics/btp666)
14. Scott LJ, Mohlke KL, Bonnycastle LL et al (2007) A genome-wide association study of type 2 diabetes in Finns detects multiple susceptibility variants. *Science* 316:1341–1345. doi:[10.1126/science.1142382](https://doi.org/10.1126/science.1142382)
15. Howie BN, Donnelly P, Marchini J (2009) A flexible and accurate genotype imputation method for the next generation of genome-wide association studies. *PLoS Genet* 5:e1000529. doi:[10.1371/journal.pgen.1000529](https://doi.org/10.1371/journal.pgen.1000529)
16. Parra G, Bradnam K, Korf I (2007) CEGMA: a pipeline to accurately annotate core genes in eukaryotic genomes. *Bioinformatics* 23:1061–1067. doi:[10.1093/bioinformatics/btm071](https://doi.org/10.1093/bioinformatics/btm071)
17. Hunt M, Kikuchi T, Sanders M et al (2013) REAPR: a universal tool for genome assembly evaluation. *Genome Biol* 14:R47. doi:[10.1186/gb-2013-14-5-r47](https://doi.org/10.1186/gb-2013-14-5-r47)
18. Hoffmann S, Otto C, Kurtz S et al (2009) Fast mapping of short sequences with mismatches, insertions and deletions using index structures. *PLoS Comput Biol* 5:e1000502. doi:[10.1371/journal.pcbi.1000502](https://doi.org/10.1371/journal.pcbi.1000502)
19. Trapnell C, Pachter L, Salzberg SL (2009) TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics* 25:1105–1111. doi:[10.1093/bioinformatics/btp120](https://doi.org/10.1093/bioinformatics/btp120)
20. Li H, Durbin R (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* 25:1754–1760. doi:[10.1093/bioinformatics/btp324](https://doi.org/10.1093/bioinformatics/btp324)
21. Yu X, Sun S (2013) Comparing a few SNP calling algorithms using low-coverage sequencing data. *BMC bioinformatics* 14:274. doi:[10.1186/1471-2105-14-274](https://doi.org/10.1186/1471-2105-14-274)
22. Farrer RA, Henk DA, MacLean D et al (2013) Using false discovery rates to benchmark SNP-callers in next-generation sequencing projects. *Sci Rep* 3:1512. doi:[10.1038/srep01512](https://doi.org/10.1038/srep01512)

Chapter 17

Finding and Characterizing Repeats in Plant Genomes

Jacques Nicolas, Pierre Peterlongo, and Sébastien Tempel

Abstract

Plant genomes contain a particularly high proportion of repeated structures of various types. This chapter proposes a guided tour of available software that can help biologists to look for these repeats and check some hypothetical models intended to characterize their structures. Since transposable elements are a major source of repeats in plants, many methods have been used or developed for this large class of sequences. They are representative of the range of tools available for other classes of repeats and we have provided a whole section on this topic as well as a selection of the main existing software. In order to better understand how they work and how repeats may be efficiently found in genomes, it is necessary to look at the technical issues involved in the large-scale search of these structures. Indeed, it may be hard to keep up with the profusion of proposals in this dynamic field and the rest of the chapter is devoted to the foundations of the search for repeats and more complex patterns. The second section introduces the key concepts that are useful for understanding the current state of the art in playing with words, applied to genomic sequences. This can be seen as the first stage of a very general approach called linguistic analysis that is interested in the analysis of natural or artificial texts. Words, the lexical level, correspond to simple repeated entities in texts or strings. In fact, biologists need to represent more complex entities where a repeat family is built on more abstract structures, including direct or inverted small repeats, motifs, composition constraints as well as ordering and distance constraints between these elementary blocks. In terms of linguistics, this corresponds to the syntactic level of a language. The last section introduces concepts and practical tools that can be used to reach this syntactic level in biological sequence analysis.

Key words Repeats, Transposon, Indexing, Algorithmics on words, Pattern matching

1 Introduction

A salient feature of eukaryotic genomes is the number of repeated sequences that they contain. Many processes contribute to this accumulation of genomic material. Even if the polyploidy speciation mechanism were not taken into account, plant genomes are particularly rich in copy events that explain the remarkable range of their size variation and may considerably increase this size. Indeed, the genome length world record, 1.5×10^{11} DNA base pairs, is currently held by a plant, *Paris japonica*, an octoploid native to subalpine regions of Japan. The aim of this chapter is to propose several methods that help identify the various repeats that populate

DNA sequences. The objective is more to collect and explain key concepts rather than to present an exhaustive and somewhat tedious study of the search for the various known repeat types, which may quickly have become obsolete.

In fact, due to its importance, we will describe in detail only the exploration of the major source of repeats, the transposable element (TE) super-class. Transposable elements often account for up to 40 % of plant genomes and, for example, account for about 60 % of *Solanum lycopersicum* and *Sorghum bicolor*, and 80 % of the *Triticum aestivum* or the *Zea mays* genome sizes. A whole section is devoted to practical methods that have been developed to extract TEs. This includes both generic tools and tools tailored for a particular class of RNA or DNA transposons. Another important class of repeats in some plants, such as *Olea europaea* [1], the tandem (or satellite) repeats, are not treated specifically, although some pointers to ab initio methods for transposable elements, such as RepeatExplorer [2], may be useful for the search of tandem repeats on reduced sets of genomic reads. We refer the interested reader to a fine review on this topic in ref. [3].

We then propose a more advanced section on the algorithmic basis of the detection of copies. This will help the reader to gain a better understanding of the technical terms often used in the description of the previous tools. All these software solutions derive much of their power from a crucial step, which finds all exact multiple occurrences of words in a sequence. We introduce the state-of-the-art data structures that are used for this task and describe how approximated repeats can be searched for efficiently from these. In all cases, we provide pointers to free available software. Next Generation Sequencing (NGS) data are tending to predominate the current technology and need to be addressed specifically. Even if NGS avoids some issues with repeat cloning in BACs, sequencers have a certain bias that can affect the search for repeats. For instance, Illumina GA sequencers have some difficulties with GGC repeats and long inverted repeats [4] and Roche 454 sequencers have a higher error rate on long A or T homopolymers [5]. Moreover, assembly is error-prone with respect to repeats, and we discuss to what extent it is possible to work directly at the read level without requiring an assembly step. The study of repeat variations is certainly the most important challenge that is currently addressed in advanced research on repeats by using NGS data, and the section ends with a small discussion on this subject.

We end the chapter with a more prospective section, which emphasizes a global and long-term approach that may be very useful in the systematic study or the discovery of particular repeat classes. It is motivated by the fact that as knowledge on repeat families grows, software solutions have to be adapted to take into account such knowledge, at an increasing cost. The linguistic approach is based on the design of some specific language for the description and search of complex chaining structures, which may

occur in nucleic acid sequences as well as in protein sequences. In this paradigm, the description of the state-of-the-art knowledge is left to the biologist acting as modeler. Each model written in the representation language can then be searched in a systematic way in the input data, using generic software, a parser, which is able to recognize the occurrences of any pattern written in the language. Of course, the more expressive languages are more desirable but this has a computational cost, which may even be beyond the reach of any computing system if the language and models are not designed carefully. A first subsection recalls the basic aspects of the theory of languages that clarify these computational limitations. A second subsection introduces some practical tools that may be experimented with for this modeling approach.

2 Detecting Transposable Elements in Plant Genomes

Many tools have been developed to search for transposable elements, forming a representative set illustrating the variety of techniques that can be applied for the search of genomic repeats in general.

The most common approach used to detect TEs is by homology to already-known TE sequences. This means that the search starts from an existing database of identified TE families and new elements are classified in the family containing the most similar sequence. In practice, this is often done on protein coding sequences (e.g., reverse transcriptase) since reaching a significant level of similarity requires a sufficient conservation level throughout evolution. In all cases, it assumes that a database of known TEs is available. A well-known generic database for eukaryotic consensus repeated sequence models is Repbase Update [6], maintained by the Genetic Information Research Institute (GIRI, Mountain View, CA). The current version, at the date of this publication (Jul. 2014), contained 8300 loci related to the Viridiplantae clade. Some more specific databases exist. In Table 1, we list a few plant transposable

Table 1
List of plant transposable element databases that cover several plant genera

Database and reference	Content	Website
mips-REdat + mips-REcat [128]	All repeats + repeat type index (82 genera)	http://mips.helmholtz-muenchen.de/plant/recat/
Plant Repeat Databases [138]		http://plantrepeats.plantbiology.msu.edu/
MASiVEDb [139]	LTR (37 species)	http://databases.bat.infospire.org/masivedb/
P-MITE [140]	MITE (41 species)	http://pmitte.hzau.edu.cn/django/mite/

element databases that cover several plant genera. Two of them cover all kinds of repeats that can appear in genomes, the most complete to date being mips-REdat.

A second approach for detecting TEs, called *structure-based* [7], tries to be less dependent on known sequences by taking advantage of a priori knowledge of characteristic features of transposon families. For instance, LTR sequences have been analyzed for more than 20 years and a number of common elements have been discovered in their architecture: a short direct repeat string marking the insertion and flanking the 5' and 3' extremities of the LTR and DNA transposons, a similar TG..CA box at each extremity, a polypurine tract about 12 bp in length, and various protein domains. All these features can serve as specific constraints enabling the detection of new elements in genomes. They usually form the basis of specific programs or scripts. We will see in the last section that this constrain solving problem may also be seen as an instance of a general pattern-matching issue, which could be treated by generic programs (parsers).

The most complex approach for detecting TEs is *ab initio*, without any assumption of the type of transposon being looked for. Typically, this is used at large scale to annotate new genomes. For this reason, *ab initio* methods make use of the most advanced data structures and string algorithms.

Of course, it is sometimes difficult to clearly state the status of a particular method, which may incorporate several steps belonging to different approaches, for instance, *ab initio* and by homology. We have created a fourth category, *pipeline*, to account for these more complex frameworks which implement a workflow combining existing software components with their own specific glue code and possibly new contributions.

Most detection methods are listed on the Bergman website.¹ Previous reviews classified the software according to the detection method [7] rather than detected transposable element superfamilies. This is a valuable approach for Bioinformatics and for whole-genome annotation and we start with a comprehensive subsection on this issue of systematic TE search in a newly sequenced genome. Since biologists are often primarily interested in certain TE superfamilies, looking at methods as a second step, we propose, in the remaining subsections, different algorithms targeting specific transposable element (TE) families and then subdividing their presentation by the method itself (*ab initio*, *homology-based*, *structured* and *pipeline* methods). Our chapter focuses on software that is currently available via code downloading, a website or by e-mail. For each TE class and each type of method, a table summarizes the corresponding list of software and provides for each a reference, a download site and installation requirements.

¹ http://bergmanlab.smith.man.ac.uk/?page_id=295

2.1 Large-Scale Search of Transposable Elements

First, we present tools that detect all TE superfamilies. These methods correspond to the majority of the TE detection tools. They generally use sophisticated algorithms to ensure an efficient search of elements and the interested reader is referred to Subheading 3 for having more details on the way it works on computers and having the possibility to compare related techniques.

2.2 All TEs; Ab Initio Methods

Available ab initio methods are listed in Table 2. They can be roughly split into two categories: methods that detect *exact repeats* of fixed size and assemble or extend them, and methods that directly detect *non-exact repeats*.

Except for Tallymer [8] and RepARK which detect a range of exact repeats, all methods in the first group detect exact repeats of length k called k -mers. These repeats are stored in various data structures offering a tradeoff between the required amount of memory and the amount of computation. These repeats are stored in various data structures offering a tradeoff between the needed amount of memory and the amount of computation. Reputer [9] and RepeatFinder [10] use the suffix tree data structure for storing and retrieving k -mers, while Tallymer [8] uses the more compact suffix array data structure. A number of methods such as WindowMasker [11] use a structure that does use the fact that elements are strings, a hash index (not to be confused with h-index...!). RepARK counts k -mers up to $k=31$, making use of the Jellyfish software [12], which is based on a multithreaded, lock-free hash index. PClouds [13] uses a bit array and a hash index. A software such as RepSeek [14] uses a lower and an upper bound for k and keeps only the maximal repeats, those that cannot be extended without losing an occurrence.

Available software solutions offer various alternatives regarding the choice of k , some more practical than others depending on the data set: it is most often a parameter whose value has to be user-provided (ReAS, Repseek, REputer and RepeatScout); it equals $\log_4(n) + 1$ for PClouds (where n is the sequence size); it is the smallest integer that satisfies the equation $n/4k < 5$ for WindowMasker. In the case of Tallymer, it is an optimal value calculated from a user-provided range fixing the minimal and maximal values. To be more precise, Tallymer calculates this value from the *uniqueness* ratio, which is the ratio of k -mers occurring exactly once relative to the total number of k -mers in the genomic sequence. Then the selected k corresponds to the least value where the ratio does not increase significantly with the increase of k (inflection point in the uniqueness ratio graph). All repeats greater than k are written out by Tallymer. RepARK assumes a Poisson distribution for the k -mers unique in the whole genome and thus determines a frequency threshold to filter significant k -mers that occur at least twice in the genome.

Unlike other methods that detect repeats on genomic sequences, ReAS and RBR work on Whole Genome Shotgun

Table 2
List of available ab initio methods detecting all types of transposable elements

Software name	Website, Download site (ftp, forge, galaxy, or github), or e-mail contact	OS	Requirements	Comments
PClouds [13]	www.evolutionarygenomics.com/ProgramsData/PClouds/PClouds.html	Linux, Mac OS X	C compiler	No help file
PILER [18]	www.drive5.com/piler/	All	MUSCLE	User's guide (website)
RBR [141]	www.i.uib.no/~ketil/bioinformatics/tools.html	All, Linux preferred	Glasgow Haskell Compiler (GHC)	README Install file
ReAS [142]	ftp.genomics.org.cn/pub/ReAS/software/	All	Perl, C compiler, 64-bit system	README Install file
RECON [19]	selab.janelia.org/recon.html	All	C compiler	README Install file. Makefile to be adapted
RepeatFinder [10]	cbeb.umd.edu/software/RepeatFinder/	All	REPuter	README Install file
RepeatScout [59]	repeatscout.bioprojects.org/	Linux, Mac OS X	Perl, Tandem Repeat Finder, RepeatMasker	README Install file
Repseek [14]	www.abi.snv.jussieu.fr/public/RepSeek/	Linux, Mac OS X	No	README Install file User's guide (Repseek_doc.pdf)
REPuter [9]	bibiserv.techfak.uni-bielefeld.de/reputer/	All	No	User's guide (website)
RepARK [116]	https://github.com/PhKoch/RepARK	All	Perl, Jellyfish, Velvet	README
Tallymer [8]	www.zbh.uni-hamburg.de/?id=211	Linux, Mac OS X	Perl, C compiler, Python, Ruby, Cairo & Pango lib. (optional), HMMER	User's guide (website)
Windowmasker [11]	ftp.ncbi.nlm.nih.gov/pub/agarwala/windowmasker	Windows, Linux	No	User's guide (website)

(WGS) and Expressed Sequence Tags (EST) data respectively. ReAS selects WGS that contains k -mers with a high copy number and divides them into 100-bp segments centered on that k -mer. ReAS clusters the WGS containing the same k -mers and creates the consensus sequence of the 100-bp segments. In the RBR method, k -mers are considered as repeats if their frequency is higher than a calculated threshold based on a binomial distribution model.

The next important feature for comparing these tools is the way they build approximated repeats from exact ones. Dynamic programming is a widely used method for this purpose, although it is not the only one and the way it is applied may vary.

Tools such as REPuter, RepSeek, and RepeatScout use k -mers as seeds and try to *extend them on both sides*. REPuter allows a fixed number of mismatches and uses a dynamic programming algorithm. For each repeat found, an E -value score is computed using the Kurtz and Myers procedure [15] which corresponds to the significance of mismatch repeats. REPuter also possesses a graphical interface to display the position of copy pairs along the genome, something that may help provide a global view of duplicated material between chromosomes. RepSeek also extends the k -mer seeds by a dynamic programming approach but accepts more errors (substitutions or indels). Finally, RepSeek calculates for each extended repeat a score based on its length and nucleotide composition. The probability for a given repeat score is computed by estimating the probability $P(S_{\text{best_repeat}} \geq S)$ that the score of the best local alignment observed between two random sequences of fixed size is larger than a given score S and is expressed as a function of the sequence length and the GC ratio. A minimum threshold score is derived from this probability above which no repeats are expected to be found in random sequences and only repeats with a score higher than this threshold are kept. RepeatScout uses a greedy algorithm to extend the exact repeat to the right and to the left and build a consensus. For each extended nucleotide, the algorithm calculates a score that is computed by summing up local similarities of each sequence with the consensus and uses an incomplete-fit penalty for sequences that partially match the consensus. The extension is stopped when the algorithm finds a locally optimal value that does not increase after a fixed number of steps (100 by default). RepARK uses a de Bruijn graph assembler for NGS data, Velvet [16], to get the repeat library.

The previous strategy can be enhanced by specific preprocessing treatments. PClouds first excludes tandem repeats. It then clusters similar k -mers (called '*P-cloud core*') and extends them according to empirical user thresholds [13].

Another approach considers that every relevant approximate repeat is made of a *mosaic of exact repeats*. RepeatFinder merges exact repeat pairs if the gap distance between the repeats is smaller than the maximal allowed gap size (a user parameter to be defined

in the command line), or if one repeat overlaps another by more than 75 %. RepeatFinder then clusters the merged repeats that have a same maximal repeat or a common hit with an E-value lower than a user-specified parameter, using WU-BLAST [17].

ReAS starts from the consensus sequence of 100-bp segments built on WGS clusters containing the same k -mers. The ReAS algorithm recursively extends the consensus sequence to the left or/and to the right if another 100-bp consensus sequence overlaps it.

Finally, WindowMasker [11] aims solely to *mask low complexity regions* and repeats in genomic sequences and it does not attempt to annotate or classify them. WindowMasker screens the sequence twice for processing repeats: the first screen computes the frequency of each k -mer (for the direct and reverse strands) and defines several thresholds corresponding to various percentiles of the empirical cumulative distribution of repeat occurrences; the second screen uses a frequency-based score and masks k -mers whose score exceeds the 99.5 % percentile or is between two masked k -mers and has a score exceeding the 99 % percentile.

In the second group of ab initio methods, no index is built and repeats are directly detected through the self-alignment of genomic sequences. These methods are thus heavily reliant on dynamic programming and their sensitivity depends on the way in which significant aligned pairs are filtered. PILER [18] and RECON [19] proceed by first aligning each sequence against itself and then, in a second step, selecting repeats that present a high copy number or a score higher than a fixed threshold.

In more detail, RECON (1) creates a graph $G(V,E)$ such that vertices V correspond to subsequences involved in WU-BLAST pairwise alignments and edges E represent overlapping vertices in a global alignment; (2) removes from this graph sequences that do not overlap a significant number of times with sequences belonging to the same cluster of sequences; (3) groups elements that belong to the same repeat family. RECON then creates a new graph $H(V',E')$ for each family such that a vertex V' corresponds to a RECON element and each edge E' corresponds to the overlap between two elements in a global alignment. A RECON repeat family is attributed to each repeat position in genomic sequences.

From the alignment, PILER defines a *pile* as a list of pairwise hits covering a contiguous region in sequences. PILER saves only hits that cluster with more than p instances (p is a user-defined parameter). An interesting original feature of PILER is that it distinguishes different categories of repeats that correspond to different pairwise hit definitions (or repeat builds) and implements a specific method for each category. Thus, PILER-DF looks for *intact transposable elements* and aligns at least three similar piles (not hits) to create a repeat, with this pile alignment avoiding generating fragmented sequences. PILER-TR searches *TEs with terminal repeats* (LTR retrotransposon) and aligns *banded* hits (i.e., hits separated by a maximum distance) to create the repeats.

AAARF (Assisted Automated Assembler of Repeat Families) is a simple Perl script that is representative of methods oriented towards the use of incomplete genomic information collected through large sets of reads. It has been tested with success on the maize genome, either with a sample of the TIGR Sanger reads (780 bp on average) or with simulated 454 reads (100 bp on average). It is most useful for trying to detect repeats from the short 454 reads, a widespread situation for biological labs. It is possible to tune a number of parameters using BLAST [21] and ClustalW. Of course, it has some limitations since it is a purely *de novo* method working on highly fragmented data and it should be used with care particularly when discriminating families of degenerated repeats, such as a series of tandem repeats or a mix of non-autonomous and autonomous transposable elements.

2.3 All TEs; Homology-Based Methods

Available *homology-based* methods are listed in Table 3. Most of them make direct use of the BLAST software [17] and a BLAST-parser. Currently, these methods used three BLAST versions: AB-BLAST [22], NCBI-BLAST [21], and PSI-BLAST [23]. The BLAST method is a famous and widely used seed-based approach that looks for the presence of a query from its k -mers and estimates a E-value for the significance of matches. Its principles are recapped in Subheading 3. For all homology-based methods, the query sequence is a library of consensus sequences from TE copies.

AB-BLAST is the new and commercial version of WU-BLAST (no longer supported) and is only free in a limited version available to academic users. The input and output files and the binary names are identical to WU-BLAST. Because there are so far no papers published and no source code, it is not possible to describe the algorithm of this new version in further detail. *NCBI-BLAST* is a widely used BLAST algorithm variation. By default, k is set to 11 for the DNA (or RNA) and to 3 for the proteins. *PSI-BLAST* is another version from the NCBI laboratory which is much more sensitive in picking up distant evolutionary relationships than a standard protein–protein BLAST [23].

A *BLAST parser* is a method that reads the BLAST hits and tries to assemble them to obtain summarized results. Since large scale analysis may produce an amount of results that can reach the amount of input sequences, such a component has a decisive role in practice. It determines the type of annotations that can be expected as a result. The language used to program the parser is an indication of its type of usage: *Censor* [24], *RepeatMasker* [25, 26], *TESeeker* [27], *TransposonPSI* [28], and *RelocaTE* [29] are written in Perl and are routines that can be included in more complex workflows; *TARGeT* [30] is written in PHP as is intended for use only via a website. Note that, independently of languages, and perhaps because it may be a tedious task for biologists, some authors

Table 3
List of available homology-based methods detecting all types of transposable elements

Software name	Website, Download site (ftp, forge, galaxy, or github), or e-mail contact	OS	Requirements	Comments
AB-BLAST (wu-) [22]	www.advbiocomp.com/blast.html			User's guide (website)
Censor [24]	www.girinst.org/censor/download.php	Linux, Mac OS X	Perl	README Install file
HMMER [45]	hmmer.janelia.org/	Linux, Mac OS X	Perl	User's guide (website)
NCBI-BLAST [21]	ftp.ncbi.nlm.nih.gov/blast/executables/blast+/	All	No	README Install file
One code to find them all [42]	doua.prabi.fr/software/one-code-to-find-them-all	All	Perl	Tutorial (zip file on website)
Process_hits [39]	processhits.sourceforge.net/	All	Perl	README Install file
REannotate [40]	www.bioinformatics.org/reannotate/index.html	All	Perl	User's guide (website)
RelocaTE [29]	github.com/srobb1/RelocaTE	All	Blat, Bowtie 1, BioPerl, SAMtools	README User's guide (website)
Repeat Masker [25]	www.repeatmasker.org/	All	Perl, Blast or HMMER or RMBlast, and Tandem Repeat Finder	README Install file
Repeat Runner [41]	www.yandell-lab.org/software/repeatrunner.html	All	Perl, Blast, and RepeatMasker	README Install file
RetroSeq [43]	github.com/tk2/RetroSeq	Linux	Perl, bedtools lib., Samtools, and Exonerate	User's guide (website)
T-lex [44]	petrov.stanford.edu/cgi-bin/Tlex_manual.html	Linux	Perl, RepeatMasker, Maq, SHRIMP2, BLAT, Phrap, and FastaGrep	User's guide (website)
TARGeT [30]	target.iplantcollaborative.org/	All	Web browser	User's guide (website)
TESeeker [27]	repository.library.nd.edu/view/27/teseeker	All	VirtualBox, BLAST, CAP3, ClustalW2, and BioPerl	User's guide (website)
Transposon-PSI [28]	transposonpsi.sourceforge.net/	All	Perl, Psi Blast	README Install file

give only a very brief account or do not write out their algorithm in the corresponding paper, something that is not favorable for the interpretation of results and rational improvement of software by the community. For example, the authors of RepeatMasker did not write a paper about it, although it is possibly the most widely used software for TE detection. *TransposonPSI* includes a library of ORFs coding for transposon proteins and uses PSI-BLAST to detect intact coding TEs and TBLAST for slightly degenerated coding TEs.

To be more precise, *TARGeT* (Tree Analysis of Related Genes and Transposons) is a webserver that does not only detect TEs in genomic sequences but also establishes a phylogeny of these elements. It first uses NCBI-BLAST to find the TE copies, aligns them with MUSCLE [31] and determines consensus with the TE family. Finally, it launches FastTree [32] to create the phylogenetic tree of the TE copies. *Censor* is a BLAST-parser that can use AB-BLAST or NCBI-BLAST. Censor first launches DUST and NSEQ to remove the micro- and mini-satellites from the genomic sequences. From the BLAST hit positions, it tries to assemble them based on the TE consensus families available in the RepBase database. Censor finally calculates for each match the new similarity score and a hit score. The software *TESeeker* detects preferentially autonomous and complete transposable elements, in three steps. In the first step, TESeeker uses NCBI-BLAST to find the partial hits of the transposable element copies, assembles them using CAP3 [33] and creates new consensus using ClustalW [34]. Hits with *E*-values higher than 1×10^{-20} are eliminated. In the second and third steps, TESeeker iterates the same method with NCBI-BLAST, CAP3, and ClustalW, but starting from the TE consensus library created during the previous step. *RelocaTE* is a list of Perl scripts aiming at the identification of given reference TE in NGS short reads (paired or unpaired). It produces the locations of TE insertions that are either polymorphic or shared between the reference and short reads. The identification is based on three tools, BLAT, Bowtie, and SAMtools. RelocaTE has been used to characterize the amplification of mPing in *Oryza sativa* [29].

Like Censor, *RepeatMasker* first detects and removes the micro- and mini-satellites, by applying Tandem Repeat Finder (TRF) [35]. RepeatMasker is very flexible and can launch many BLAST-like software solutions: cross-match [36], NCBI-BLAST [21], AB-BLAST [22], or Decypher [37]. The NCBI laboratory has made available a special BLAST version for RepeatMasker called RM-BLAST [38]. RepeatMasker identifies the TE superfamily of the fragmented BLAST-like hits and assembles them using methods tailored to each superfamily.

RepeatMasker is a base component used in many applications. There are (at least) four other methods that read the RepeatMasker hit outputs and reassemble them: Process_hits [39], REannotate [40], RepeatRunner [41], and “One code to find them all” [42].

Process_hits, a Perl script that can also read a BLAST output, processes a refined output using various formats depending on the user parameters. *REannotate* defragments RepeatMasker outputs following certain rules: the two fragmented hits have the same orientation; the distance between the first and the last fragment must not exceed a user-specified parameter (default 40 kb), and the overlap between two fragmented hits must exceed 10 % of the reference length sequence. *RepeatRunner*, written in Perl, mainly detects the autonomous transposable elements (those that contain an ORF). RepeatRunner first launches BLASTX [22] and RepeatMasker, merges the results of both in a XML-based output and eventually masks the repeats in the input genomic sequence (like RepeatMasker). “*One code to find them all*” assembles RepeatMasker hits into complete copies, retrieve corresponding TE sequences and flanking sequences, and compute summary statistics for each TE family.

We end with three other methods that do not use BLAST-like output to detect transposable elements but can nevertheless be related to the homology-based methods, RetroSeq, T-lex, and HMMER.

RetroSeq [43] and *T-lex* [44] detect TEs in next-generation sequencing (NGS) reads and are described for this reason in Subheading 3.3. *HMMER* [45] is a widely used pattern-matching and discovery software package which creates and searches profile Hidden Markov Models (profile HMMs) against a sequence. It was not specifically designed for TE detection. Three steps are involved in looking for a transposable element belonging to a fixed family using HMMER. One must first align the known copies of the TE family (HMMalign) and then create the HMM profile of this family (HMMbuild and HMMcalibrate). The final step consists in scanning the genomic sequence with the HMM profile as a model (HMMsearch).

2.4 All TEs; Structured Methods

To our knowledge, three *structured* methods have been applied to detect transposable elements, SMaRTFinder and SMOTIF for the Copia retrotransposons in *A. thaliana* (also with SMaRTFinder) and STAN used for Helitrons in *A. thaliana*. This is the reason for citing them here although these methods are not dedicated to the recognition of transposable elements. The corresponding tools are listed in Table 4. They are generic and can be used to detect any biological pattern. This is the subject of Subheading 4.2, where more complete parsers such as Vmatch are described.

Apart from SMOTIF [46], SMaRTFinder [47], and Stan [48] use the suffix-tree data structure. SMaRTFinder and SMOTIF first detect all positions of each element of the motif and join the positions that satisfy the distance between the elements of the motif. STAN, on the other hand, creates a list of the possible sequences from the user motif and looks for each of them. Only STAN can

Table 4

List of available structure-based methods detecting all types of transposable elements

Software name	Website, Download site (ftp, forge, galaxy, or github), or e-mail contact	OS	Requirements	Comments
SMaRTFinder [47]	services.appliedgenomics.org/software/smartfinder/	All	C compiler	README Install file
SMOTIF [46]	www.cs.rpi.edu/~zaki/software/sMotif/	All	C compiler	README Install file
Logol [136]	logol.genouest.org/	Linux, Mac OS X	Ruby	User's guide (website)

detect motifs with substitution errors and non-fixed gaps. STAN is no longer maintained and a more expressive tool, Logol, can be used instead. It is also described in Subhedding 4.2.

2.5 All TEs; Pipeline Methods

We end this section with the most elaborate tools, which chain several methods to enhance the repeat annotations and are listed in Table 5. Apart from DAWGPAWS [49], which uses the three kinds of method, RepeatModeler [25], REPET [50], TriAnnot [51], and RISCO [52] use de novo and *homology-based* methods to detect TEs. All these programs are meta-tools: they launch other software and assemble and rewrite their results. Among TE detection software, only RepeatMasker [26] is used by all these pipelines.

In more detail, *DAWGPAWS*, written in PERL, launches LTR_STRUCT [53], LTR_seq [54], LTR_FINDER [55], FINDMITE [56], Find_LTR [57], TRF [35], Repseek [14], RepeatMasker, HMMER, TE_Nest [58], and BLAST. *DAWGPAWS* assembles and rewrites the output of these tools. After removing the tandem repeats with TRF [35], *RepeatModeler*, also written in PERL, uses first RECON [19] and RepeatScout [59]. Finally, RepeatModeler launches after RepeatMasker with a library of consensus sequences to identify and assembles the previously detected hits. *REPET* is a sophisticated method composed of two main pipelines. The first pipeline (de novo) compares the genome against itself using BLASTER (a BLAST-like method written by the authors [60]). Then, hits are clustered with RECON, PILER [18] and GROUPE (also written by these authors [60]). A consensus sequence is created by multiple alignment [MAFFT [61] and MAP [62]] and then classified. The second pipeline (*homology-based*) looks for repeats using a library of sequences, for example the created consensus, with RepeatMasker, CENSOR [24] or BLASTER [60]. This pipeline also uses TRF and mreps

Table 5
List of available pipeline methods detecting all types of transposable elements

Software name	Website, Download site (ftp, forge, galaxy, or github), or e-mail contact	OS	Requirements	Comments
DAWGPAWS [49]	dawgpaws.sourceforge.net/	Linux	Perl, emacs, Apollo, Blast, Cross_match, EuGène, find_ltr, FINDMITE, FGENESH, GeneID, GeneMarkHMM, Genscan, HMMER, LTR_FINDER, LTR_Seq, LTR_Struc, RepeatMasker	User's guide (website)
RepeatExplorer [2]	galaxy.umbr.cas.cz:8080/	Linux	Perl, R, Python, ImageMagick, Blast, RepeatMasker, Muscle, Fasty36	User's guide (website)
RepeatModeler [25]	www.repeatmasker.org/RepeatModeler.html	Linux, Mac OS X	Perl, RepeatMasker, RECON, RepeatScout, Tandem Repeat Finder, and RMBlast or Ab-Blast	README Install file User's guide
REPET [50]	urgi.versailles.inra.fr/Tools/REPET	Linux, Mac OS X	C compiler	User's guide (website)
TriAnnot [51]	wheat-urgi.versailles.inra.fr/Tools/Triannot-Pipeline	All	Web browser	User's guide (website)
RISCI [52]	www.ccmb.res.in/www.rakeshmishra/tools.html	Linux	Perl, EMBOSS, Blast, RepeatMasker	README

[63] to detect micro- and mini-satellites. *TriAnnot* [51] is an annotation pipeline which first detects transposable elements and then predicts the other genetic elements. It launches BLASTX [21], RepeatMasker, TEAnnot [50] (*homology-based*), and Tallymer [8] (ab initio method). Finally, *RISCI* (Repeat Induced Sequence Changes Identifier) is a set of Perl scripts specialized in the comparative genomics of transposons. Starting from a reference genome and comparative genomes, it is able to infer intra-species and inter-species structural variations introduced by transposons (e.g., Target Site duplication, inversion and truncation of repeat sequence or post insertion modifications like disruption). This pipeline uses RepeatMasker, Blast, and the EMBOSS module and the GenBank annotation file if made available.

2.6 LTR Retrotransposons

Concerning the search and analysis of specific transposable elements, LTR retrotransposons (LTRR) form a large family and offer a rich structure that justifies the existence of several dedicated tools. An overview of the most interesting ones is provided in Table 6.

2.7 Homology-Based Methods

LTR_MINER [64], written in Perl, is the only homology-based method specialized in LTR retrotransposon detection. It launches first RepeatMasker and WU-BLAST, and then assembles the multiple hits of the Long Terminal Repeat (LTR) (e.g., the extremities of the retrotransposon) under some constraints: a maximal distance between hits (550 bp), a common orientation, a same LTRR family, and a combined length no larger than a complete LTRR. *LTR_MINER* also gives information about the probable age of the LTR retrotransposon insertion.

2.8 Structure-Based Methods

LTR_FINDER [55] and *LTRharvest* [65] detect only full length LTR retrotransposons while *LTR_STRUC* [53] and *MGEScan-LTR* [57] detect all LTRs. Aside from *LTR_STRUC*, *LTR_FINDER*, *LTRharvest*, and *MGEScan-LTR* use the suffix-array data structure to find the exact maximal repeats in the genome, and extend/merge them into non-exact repeats. The three methods look also for LTRR signals such as Target Site Duplication (TSD) and PBS and PPT retrotransposon motifs.

In more detail, *LTR_FINDER* [55] is a web server that merges exact repeat pairs with a similarity score higher than a minimum extension threshold. *LTR_FINDER* combines dynamic programming (Smith-Waterman algorithm) and the search of structured motifs like the TG..CA box and TSD, in order to adjust LTRR candidate extremities. Next, *LTR_FINDER* looks for LTRR motifs such as the PBS and PPT signals (by aligning the LTR retrotransposon candidate with the 3' tail of tRNAs) or LTRR protein motifs (using ScanProsite on protein domains such as IN and RH). *LTRharvest* first builds the suffix array of the genomic sequence

Table 6
List of available software for the search of Long Terminal Repeat Retrotransposons (LTRR)

Software name	Method	Website, Download site (ftp, forge, galaxy, or github), or e-mail contact	OS	Requirements	Comments
LTR_MINER [64]	Homol.	genomebiology.com/content/supplementary/gb-2004-5-10-r79-s7.pl	All	Perl	No help file
LTR_Finder [55]	Struct.	tlife.fudan.edu.cn/ltr_finder/	Linux (standalone)	No	User's guide (website)
LTR_STRUC [53]	Struct.	www.mcdonaldlab.biology.gatech.edu/ltr_struct.htm	Windows	No	No help file
LTRharvest [65]	Struct.	www.zbh.uni-hamburg.de/forschung/genominformatik/software/ltrharvest.html	Linux, Mac OS X	Perl, C compiler, Python, GenomeTools	README Install file User's guide (website)
MGEScan-LTR [57]	Struct.	darwin.informatics.indiana.edu/cgi-bin/evolution/daphnia_ltr.pl	Linux, Mac OS X	Perl, C compil. Tandem Repeat Finder, HMMER, EMBOSS	README Install file User's guide (website)
MASIVE [67]	Pipel.	tools.bat.infospire.org/masive/	Linux, Mac OS X	LTRharvest, Vmatch, Wise2 and MAFFT	README Install file

using GenomeTools [66] and stores all maximal repeats that are longer than a user-defined threshold. Optionally, LTRharvest looks for motifs such as the TSDs, which can be derived from the maximal repeat occurrences, and the palindromic LTR motif, which corresponds to the dinucleotide palindrome in the LTR extremities (often TG with CA). Finally, LTRharvest checks candidates that have the TSD site, together with some constraints on the size and similarity of the two LTRs and the distance between them. *LTR_STRUC*, written in C++, first identifies similar pairs subject to a set of constraints: common matching size (larger than 40 bp), similarity (higher than 70 %), and distance (lower than a user-defined parameter value). In a second step, *LTR_STRUC* tries to extend from the initial pair to a second pair in the 3' direction, then the 5' direction. The extension proceeds by looking in neighboring regions of fixed size (100 bp) for a largest match pair that can be aligned at similar distances from the previous ones and greedily produces the alignment of the whole region by filling the gap by largest matches. This extension process continues until the similarity falls consistently below the 70 % threshold. A last step determines progressively the exact termini of the LTR retrotransposon by calculating the number of matches in a sliding window. Finally, *MGEScan-LTR* [57], written in C++ and Perl, uses an algorithm similar to *LTR_seq* [54] (not available) and *LTR_STRUC*. In a first step, *MGEScan-LTR* finds all maximal repeat pairs longer than 40 bp and within a range of distances (between 1000 and 20,000 bp). Two exact pairs are merged if they are close (less than 20 bp) and share a similarity greater than 80 %. In a second step, the method scans the ORFs inside the LTR candidates using HMMER [45], and removes the candidates that match with DNA transposons. In the third step, *MGEScan-LTR* looks for solo LTRs. It clusters the LTR discovered previously and aligns them to create a profile HMM of each cluster. HMMsearch, a procedure of HMMER, is used to discover these solo LTR retrotransposons.

2.9 Pipeline Methods

MASiVE [67], written in Perl, is typical of the methods based on carefully designed specific models: it only detects full autonomous LTR of the plant-specific Sireviruses and takes advantage of the highly conserved motifs they share for a sensitive and accurate search. It uses Vmatch [68]—see Subheading 4.2—to detect clusters of Sirevirus-specific PPT motifs, and LTRharvest [65]—see Subheading 2.6—to detect complete LTR retrotransposons. Only candidates that possess both hits, the right PBS site, an admissible distance between the different elements and include LTRS and an internal sequence longer than 500 bp are retained. Wise2 [69]—see Subheading 2.7—is used to detect the reverse transcriptase (RT) present in almost intact LTR retrotransposon.

2.10 Non-LTR Retrotransposons

The two methods that detect non-LTR retrotransposons first launch homology-based tools, via Perl language scripts, and also look for non-LTR motifs to confirm the TE candidates. They are listed in Table 7.

MGEScan-nonLTR [70] uses dedicated HMM profiles and the pHMM module from the HMMER package [45] to detect autonomous non-LTR retrotransposons. These profiles correspond to the apurinic/apyrimidinic endonuclease APE, the linker and the Reverse Transcriptase protein domains. *MGEScan-nonLTR* then classifies the candidates into one of the 12 known non-LTR clades.

RTAnalyzer [71] launches BLAST to detect all non-LTR matches (autonomous and non-autonomous). From the hits, *RTAnalyzer* launches *Matcher* [72], which determines precisely the 5' extremity of the non-LTR retrotransposon. From the corresponding 5' Target Site Duplication (TSD), the algorithm extracts the 3' TSD. Finally, *RTAnalyzer* determines the polyA tail, calculates a score from all these motifs and saves the non-LTR candidates that have a score higher than a threshold.

2.11 DNA Transposons

DNA transposons are made of a transposase gene flanked by Terminal Inverted Repeats (TIRs) that make structure-based methods the best adapted for an efficient search. The main difference between methods is the way in which TIRs are selected. All methods are listed in Table 8.

2.12 Homology-Based Methods

To our knowledge, there exists only one method specialized in homology-based detection of DNA transposons : *TRANSPO* [73]. It implements a fast bit-vector dynamic programming algorithm [74] that finds the position of all matches similar to a given sequence in a library. The set of matches are then clustered using the SPAT program [75].

Table 7

List of available homology-based methods detecting non-LTR retrotransposons

Software name	Website, Download site (ftp, forge, galaxy, or github), or e-mail contact	OS	Requirements	Comments
MGEScan-nonLTR [70]	darwin.informatics.indiana.edu/cgi-bin/evolution/nonltr/nonltr.pl	Linux, Mac OS X	Perl, C compiler, HMMER, EMBOSS package	README Install file
RTAnalyzer [71]	www.riboclub.org/cgi-bin/RTAnalyzer/index.pl	All	Perl, Internet connexion	No help file

Table 8
List of available software for the search of DNA transposons

Software name	Method	Website, Download site (ftp, forge, galaxy, or github), or e-mail contact	OS	Requirements	Comments
TRANSPO [73]	Homol.	algen.lsi.upc.es/receca/search/transpo/transpo.html	Windows (standalone)	cygwin1.dll (standalone)	No help file
IRF [76]	Struct.	tandem.bu.edu/irf/irf.download.html	All	No	No help file
RSPB [78]	Struct.	pmitc.hzau.edu.cn/MITE/tool	Linux	Blast, Muscle, Mdst, Perl, RepeatMasker	Readme (http://122.205.95.39/media/MITE/tools/RSPB_Readme.txt)
MITE-Hunter [79]	Struct.	target.ipplantcollaborative.org/mite_hunter.html	All	Perl, Blast, Muscle, mDust	Readme Install file User's guide (website)
MITE Digger [80]	Struct.	labs.csb.utoronto.ca/yang/MITEDigger	Windows	Perl	Readme + Rice database
MUST [77]	Struct.	csbl1.bmb.uga.edu/fzhou/MUST/	Web server	No	No help file

2.13 Structure-Based Methods

Five methods use the palindromic structure of their 5' and 3' extremities to detect DNA transposons: Inverted Repeats Finder (IRF) [76], MITE Uncovering SysTem (MUST) [77], Repetitive Sequence with Precise Boundaries (RSPB) [78], MITE-HUNTER [79], and MITE Digger [80]. They all look first for the Terminal Inverted Repeat (TIR), and all but IRF also look for the TSDs that flank the TIR, created during the insertion of the element by a staggered cut in the target DNA.

In more detail, IRF searches TIR candidates that contain exact short inverted repeats (4–7 nt) that do not overlap, then extends them, calculates an alignment score for a larger palindromic arrangement and saves the candidates that obtain values higher than the thresholds for similarity (70 %) and length (25 bp). *MUST* detects the TIR associated with the TSD (the minimal and the maximal length of TIR and TSD are user-defined parameters) from a sliding window (up to 500 bp). Candidates with a score higher than a similarity threshold are conserved. *MUST* then clusters them using the MCL algorithm [81] and writes out the MITE families that contain at least three occurrences. *RPBS* is a series of Perl scripts using a Blastn-based approach. It seems to have been essentially applied to non-autonomous transposons of the Miniature Inverted Transposable Elements family (MITE), but the core of its approach could be applied to a broader set of families. The principle is to build clusters of repeats (at least five elements, less than 1500 bp) sharing high similarity ($Evalue < 10^{-15}$) and having precise boundaries (maximum 5 bp variation and dissimilar 100-bp flanking sequences <50 % identity). This software is known to be resource-intensive (several days of computations for large genomes). *MITE-HUNTER* and *MITE Digger* have been designed exclusively to detect MITEs. In *MITE-HUNTER*, MITE candidates (with their flanking regions) are compared with each other by BLAST [21], and MITE candidates that have similar flanking regions are considered to be part of a larger repeat element and are removed. *MITE-HUNTER* clusters the remaining candidates and defines a representative element (exemplar) for each cluster (an element that has the greatest similarity with all other elements). BLAST is used to detect its homologs in the input sequences. Candidates that have many copies are then aligned. Homologs such that flanking regions of the MITE sequences are similar (>60 %) for the majority of occurrences are assumed to be false positive and discarded. The TSD is predicted again for the remaining candidates for better accuracy. The program creates a consensus sequence from the clustered homologs, compares all consensuses (using BLAST) and clusters them again in order to reduce the number of clusters. *MITE Digger* is another BLAST-based program designed to scale large genomes by making use of the fact that a MITE family typically contains several hundred highly similar copies that are scattered all over the genome. Rather

Table 9
Software looking for Helitrons

Software name	Website, Download site (ftp, forge, galaxy, or github), or e-mail contact	OS	Requirements	Comments
HelSearch [82]	helsearch.sourceforge.net/	All	Perl, Blast, ClustalW	README Install file

than repeating for each copy TIR and TSD signature identification, screening, multiple sequence alignment and clustering, the search is first focused on a small portion of the genome and a possible family is filtered out as soon as the number of found copies reaches a threshold. For instance, the probability is very low (1 %) of missing 50 copies in 8 % of a genome database. Low-complexity regions and hits with very similar flanking regions are discarded from sequences and the process is iterated.

2.14 Helitrons

There seems to be only one method to date specializing in helitron search: HelSearch [82] (*see* Table 9). This is based on the search for a helitron signature in the 5' and 3' extremities. HelSearch looks for the *CT[AG][AG]T* motif (3' termini) of the helitrons and the small hairpin structure (a 3' sub-termini with a GC-rich sequence) calculated using UNAFOLD [83]. It then determines the 5' end (which contains the dinucleotide *TC*) from the multiple alignment of the potential hits. Finally, HelSearch classifies helitrons based on their 3' end similarity and uses BLAST to detect the fragmented helitrons in the input sequences.

3 Efficient Search of Repeats in Genomic Sequences

Some common background appears in all the tools we have seen so far: in most cases, even if it seems a little remote from reality, programs are looking first for *exact* repeats, often referred to as anchor points or *seeds*. The real, approximated repeats are obtained by extending the search further from these seeds, by introducing elementary operations enabling the detection of copies with distant contents. Looking for exact and approximated repeats is part of the rich domain of string algorithms. In the most general setting, detecting repeats in genomic sequences, or in any kind of textual sequence, is organized on the basis of the detection of particular words (the queries) in a text (the bank). This setting appears clearly when processing alignments in homology-based methods or looking for particular motifs in structure-based methods.

There are basically two ways of performing such detection.

In the first case, the text is not preprocessed and the *query is searched on the fly* in the text. For instance, this is the case when using the well-known “Ctrl-F” shortcut key or search function on any piece of software dealing with texts (text editors, web browsers, etc.). Beginning in the early 1970s, there are strong and interesting theories about fast detection on the fly of queries in texts. This is known as *string-matching*. The naive way to achieve string-matching is to compare at each position in the text the presence of each character of the query, starting from this position. Obviously, if n is the size of the text and m is the size of the query, this would require $n \cdot m$ comparisons. Thinking of large texts (several gigabytes for a genome), and even for relatively small queries (say 1000 bp), this quickly becomes an issue. It is possible to decrease this complexity for a linear behavior, although it will always depend of the size of the bank. We will not address the topic in this chapter but the interested reader can refer to the excellent book from Charras & Lecroq [84], which explores string-matching algorithms exhaustively and in a didactic way. A more general approach, pattern-matching, is discussed in the last section (linguistic analysis).

Given their huge volume, the treatment of genomic sequences is generally based on the second way of performing text processing: the *text is preprocessed by indexation techniques*. In the previous section, many technical terms have been used, such as suffix trees and suffix arrays. These are structures developed for fast indexation and searching in texts. Research in this area has been fostered by developments in genomics and in Internet content querying and we propose a quick but up-to-date review of the key results achieved to date. We focus on indexing techniques which open the way both to fast queries and to repeat detection.

3.1 The Art of Indexing

Imagine what would happen without indexation while looking, for example, for a word (query) on the Internet (bank). Your favorite search engine would open and then read all words of all pages of the whole web in order to detect the presence/absence of your query in each page. Estimated to a few tens or hundreds of billions of webpages, querying the web would simply be impossible, whatever the computation resources available. The same would be the case if the bank is a large set of genomic sequences, such as GenBank. For an efficient search, indexes are necessary. An index is a mechanism that can answer the question: “where does this word occur in the bank?” within an amount of time that *does not depend on the size of the bank*. To be precise, indexes are built once for a given bank (and within a time proportional to the size of the bank) and then the time it takes to perform every query in the bank depends only on the query size. Happily, indexes also have “side effects” which are valuable in the biological context, such as the efficient detection of repeats inside or between genomes.

The main idea is to use *data-structures* to organize and structure the information initially present in the banks. A classic and well-known structure is the dictionary: searching for a word in a paper dictionary does not require reading the entire book, as words are organized in lexicographic order. Similarly, the data-structures used for indexing texts are computational objects having the same advantages as classic dictionaries: they enable a portion of a sequence to be searched quickly by avoiding all portions where this query could not occur. However, while genomes can easily be seen as particular texts, the notion of “word” is not natural in this context since no general delimiter exists. This is why indexes used in this context are designed to answer questions regarding *any word* in the bank. For example, consider a sequence $S=ATGCGCAGTTTAT$ as a bank, and a query $P=GCGC$: “does P occur somewhere in S , and, if yes, where?” In this example the answer would be “ P occurs at position 3 in S ”. From the point of view of strings, positions are associated to *suffixes* of the text, that is, words starting at this position and ending at the end of the text. For instance, position 3 in S is associated with the word $GCGCAGTTTAT$ (and P is a *prefix* of this word, that is, it is placed at its beginning). There are a number of indexes primarily powered by this important notion of the suffix: automaton, hash tables, suffix trees or suffix arrays. All of these data-structures are designed to answer at least the fundamental question “where does P occur in S ”, but each has some specific skills either in terms of additional possibilities or in terms of performance (time efficiency and memory footprint).

Here we propose focusing on the most well-known and most widely used data-structures: *suffix trees*, an “historical” data-structure which is no longer extensively applied in practice but which, in addition to useful educational and theoretical properties, also affords the most and the most flexible functionalities; *suffix arrays*, an efficient and elegant way to index large datasets and answer additional useful questions such as those related to exact repeats in a bank; and, lastly, the *Burrows-Wheeler transform* on which the compressed *FM-index* is based and which is a development that makes it possible to build compressed suffix arrays. Note that in all cases, an efficient program code appears quite short but is in fact very tricky and definitely not within a standard programmer’s reach: use well-written libraries!

3.2 Suffix Trees

Suffix trees are represented by a classification tree structure that clusters the suffixes of the indexed text. All suffixes having a common prefix are clustered in the same class under a common internal node that has this prefix as a label. Any suffix can be read along exactly one path from the unique root of the tree to one of its leaves and conversely, any path from the root to a leaf corresponds to a unique suffix of the text or, equivalently, to a position in this text. Figure 1 shows an example of the suffix tree of $S=ATTTAATAAC$. The suffix at position 8 in the text, AAC, can

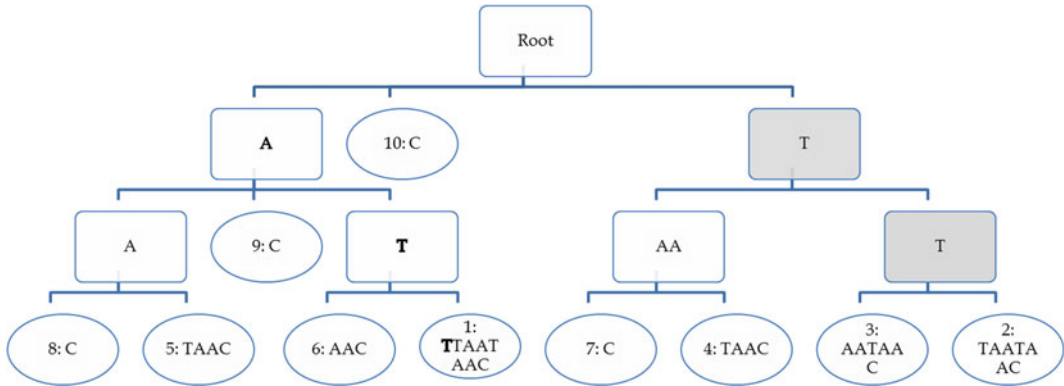


Fig. 1 Suffix-tree representation for the text ATTTAATAAC. The *top square* is the root of the tree. *Circles* are the leaves, and *other squares* are called “internal nodes”

be read from the root to the leftmost leaf labeled 8:C by collecting all the words along the path, A, A, and C.

There are strong theories and beautiful algorithms for constructing suffix trees quickly and with the lowest memory requirements. The most famous include the Weiner [85] and later the Ukkonen [86] algorithms which both propose a way of constructing the suffix tree of a sequence composed of n characters with a number of operations and a memory requirement proportional to n . To be precise, storing the suffix tree for a genomic sequence whose alphabet is limited to letters A, C, G and T requires in the worst case 20 bytes per indexed character in optimized applications [87]. Indexing the human genome therefore requires 61.47 GB of memory. Even if the actual average amount of memory required is rather around 13 bytes per nucleic acid, storing the suffix tree puts a strain on the main memory for large-scale applications. Applied, for instance, to indexing an Illumina run composed of 100 million reads of length 100 would require 130 GB of memory. In order to save space, it is necessary to shift the space/time tradeoff in the program. One option is to save the tree to hard disk rather than the main memory but this becomes much slower to access. The most recent implementations of suffix trees have used compressed structures, to the cost of slightly slower access [88]. The corresponding program is available in C++ on the website [89].

The suffix tree makes it possible to answer a query in an amount of time proportional to the length of the query. Indeed, any word P read in the tree of an indexed sequence S from its root to any one of its leaves corresponds to a word in S . For instance, searching for the word *ATT* in the suffix tree of $S=ATTTAATAAC$ can be read in bold in Fig. 1, leading to a leaf labeled 1: **TTAATAAC** meaning that the word *ATT* exists at starting position 1 in S .

Fortunately, the suffix tree offers many possibilities other than answering such simple queries. For instance, it is of great interest for detecting exact repeats. Indeed each *internal node* of the tree

(squares in Fig. 1) determines a word which has several occurrences. This is the case for the word *TT* in our example (path with grey rectangles in the Figure): the corresponding cluster contains leaves labeled 3 and 2, meaning that there are two occurrences of *TT* in *S*, starting at positions 2 and 3. Moreover, simple algorithms can exploit the suffix tree to find easily distinct kinds of repeats: those having the maximal number of occurrences, the longest, those that are *maximal* in their length (adding a new character to the repeat would discard some occurrences), etc. The suffix tree can be applied to more than one unique sequence, in this case it is called a *generalized* suffix tree. The generalized suffix tree is useful in genomics as it affords, for example, the possibility of easily determining the words that are copied between different sequences or chromosomes.

3.3 Suffix Arrays

The suffix array appeared in the early 1990s with the Manber and Myers paper [90]. This data-structure is much simpler than the suffix tree, requires less memory space and answers the same range of questions. It is constructed by first sorting all suffixes of the indexed text and by storing the obtained set in alphabetic order, remembering the obtained starting positions.

For instance, the set of suffixes of *S*=ATTTAATAAC is made of ATTTAATAAC (starting position 1), TTTAATAAC (pos. 2), TTAATAAC (pos. 3), TAATAAC (pos. 4), AATAAC (pos. 5), ATAAC (pos. 6), TAAC (pos. 7), AAC (pos. 8), AC (pos. 9), and C (pos. 10). Their alphabetic order is: AAC, AATAAC, AC, ATAAC, ATTTAATAAC, C, TAAC, TAATAAC, TTAATAAC, and TTTAATAAC. Let us represent this set of suffixes in a table:

Position	Suffix
8	AAC
5	AATAAC
9	AC
6	ATAAC
1	ATTTAATAAC
10	C
7	TAAC
4	TAATAAC
3	TTAATAAC
2	TTTAATAAC

The suffix array may be reduced to the *Position* column of the previous table (the suffixes are represented to promote better

understanding of the structure but are never stored in the computer memory). This sole column may appear rather minimalist, but this simple piece of information and the original indexed sequence are sufficient for efficiently answering a query. In its simplest form, it can be found by applying a dichotomic procedure: (1) look for the word at the middle position of the array. If the query occurs at this position the search stops, or (2) if the query is alphabetically smaller than the suffix starting at this position, recursively apply the procedure to the upper part of the array, or apply it to the lower array.

For instance, searching the query $Q=AC$ in the previously indexed text $S=ATTTAATAAC$ would entail the following steps. First check the middle of the array. In this example, it corresponds to the suffix at position 1, that is, $ATTTAATAAC$. As it is alphabetically larger than the query $Q=AC$, the process restarts with the array limited to:

Position	Suffix
8	<i>AAC</i>
5	<i>AATAAC</i>
9	<i>AC</i>
6	<i>ATAAC</i>

Consider the middle line of this array: it corresponds to the suffix starting position 5, $AATAAC$, which is alphabetically smaller than the query $Q=AC$. Thus the process restarts with the lower part of the array. In this case, position 9, where the suffix AC starts, corresponds to the searched query. With this simple procedure, at worse, the number of steps to be performed is $\log(\text{size of the array}) = \log(\text{size of the indexed sequence})$. The query time can be further improved by the use of a new column called the LCP, which provides the length of the “Longest Common Prefix” between two consecutive lines in the suffix array. For instance, if 0 is an LCP default value for the first suffix of the array, the two lines of the previous suffix array would become:

Position	LCP	Suffix
9	0	<i>AAC</i>
6	2	<i>AATAAC</i>

The value 2 indicates that AAC and $AATAAC$ start with 2 common characters. The LCP somehow enables retrieval of the suffix tree topology, and then enables answering questions related to repeats. In the example, $LCP = 2$ indicates that a repeat of size 2 exists in the indexed text, at positions 9 and 6 given by *Position*.

The suffix array (together with the LCP array) requires (for classic genomic analyses) 5 bytes per indexed character. Thus, indexing a 3.3 billion-character human genome with a suffix array and its LCP would require 15.37 GB. The indexation of a hundred million reads of length 100 would require 46.57GB of memory. A good C++ code is available in the appendix of the paper [91] (algorithm DC3) and two other C programs that are among the most advanced implementations to date are given in the appendix of the paper [92] (algorithms SA-IS and SA-DS). We mention the recent paper [93], a didactic presentation of the SA-IS algorithm for Bioinformatics specialists. Regarding access to an operational code in Java, please refer to [94]. A recent development in using suffix arrays to look for repeats (finding the largest substring common to a set of sequences and finding maximal repeats exclusive of a sequence with respect to another set of sequences) is described in ref. [95]. The C code can be downloaded from ref. [96]. In fact, genomic repeats may be a source of inefficiency for certain implementations and care is required when choosing a program. A far more general use of the suffix array data-structure is described in the next section with the software Vmatch [68].

3.4 FM-Index and Burrows-Wheeler Transform

The Burrows-Wheeler transform [97], inspired by the suffix array, was the starting point of the FM-index [98, 99], a new powerful indexing approach that is now widely used in many fields, in particular in computational biology. The Burrows-Wheeler transform (*BWT*) is a permutation of characters in a sequence that is easy to understand from a suffix array. Using our previous example, *ATTTAATAAC*, it is the sequence *TTAACAATTA*, which can be displayed next to the suffix array as follows:

Position	Suffix	BWT
8	<i>AAC</i>	<i>T</i>
5	<i>AATAAC</i>	<i>T</i>
9	<i>AC</i>	<i>A</i>
6	<i>ATAAC</i>	<i>A</i>
1	<i>ATTTAATAAC</i>	<i>C</i>
10	<i>C</i>	<i>A</i>
7	<i>TAAC</i>	<i>A</i>
4	<i>TAATAAC</i>	<i>T</i>
3	<i>TTAATAAC</i>	<i>T</i>
2	<i>TTTAATAAC</i>	<i>A</i>

For each position in the suffix array, the BWT corresponds to the letter located just before this position in the sequence (or the last letter for position 1). For instance, the letter indicated in the first line is ‘T’ as it is the letter at position $8 - 1 = 7$ in the text. The *BWT* has astonishing properties that we review here briefly. Interested readers can refer to [97] for further algorithmic details.

A first important property of the BWT is that this permutation (*TTAACAATTA* in our example) is reversible: it is sufficient to retrieve the indexed sequence (*ATTTAATAAC*). This is what is called a *self-index*, i.e., it is not necessary to keep the indexed sequence in the memory, it is contained in the index.

A second appreciable property of the BWT is that it is highly compressible. Indeed this letter organization tends to create stretches of letters. For instance, consider the sequence “treat pea repeats” (spaces added to help reading). Its BWT sequence is “eeeepprrtttetataasa”. This is well-suited for compression algorithms (e.g., “eeee” could be rewritten as “4e”).

In 2000, Ferragina and Manzini refined this compression approach to make it capable of answering queries in a text, leading to the FM-index (its full name, “Full-text index in Minute space” does not really help its understanding). To do this, they added a few pieces of information to the *BWT* while keeping the data structure extremely light. We cannot enter into the details of this structure, but suffice to know that it can be considered as a kind of compressed suffix array requiring very little memory space. For instance, the human genome could be indexed with this approach using 1.23 GB, and one million reads of length 100 would require 3.73 GB of memory. A good source for codes on the FM index and suffix array indexes is the Pizza&Chili reference site [100]. The FM-index supports the following operations which generally use a tiny portion of the compressed file:

- *Locate* finds the position in the text of an occurrence of the query, in $O(\log^c n)$ time, where c is a constant chosen at the time the FM-index is built.
- *Count* computes the number of occurrences of the query, in time proportional to its length.
- *Extract* returns the sequence of a given length starting at a given position in the text.
- *Display* outputs for a given length L the L characters on each side of the occurrences of the query in the text.

3.5 Hash Tables Versus Burrows- Wheeler Transform

Another frequently employed indexing data structure is the hash table. The hash table concept stands on a very simple idea. For indexing a set of names for instance, each name is converted into an address (called a hash value) using a function that is easy to compute. For instance, using a hash function that considers each

substring as a number in base 11 (each letter being coded by an ASCII number), “AMY” is given the address $8801 = 65 \times 11^2 + 77 \times 11^1 + 89 \times 11^0$ and “BOB” is given the address $8921 = 66 \times 11^2 + 79 \times 11^1 + 66 \times 11^0$. In an initially empty array (called the hash table), “AMY” is stored at position 8801 and “BOB” at position 8921. As it is important to save as much space as possible, it happens that two different names get the same hash value. For instance, imagine now we add the name “AYM” that also has the hash value 8921. This causes a collision at position 8921 that already contains the name “BOB”. There are several ways to manage collision, either by computing a new hash value for “AYM” (open addressing) or by storing at each position a list of name (“BOB” and “AYM”). In case of too many collisions, the hash table is overloaded and may be resized. This operation is expensive as it may reorganize all the already stored items.

Querying a hash table is fairly similar to what is done by the indexing algorithm. The hash value of the query is computed, and the content of the hash table at this position is visited in order to check the presence/absence of the queried object. Depending on the strategy used to manage collisions, either all the entries present in the list at this position have to be checked or new hash functions have to be computed for the query until it is found (match) or an empty position is reached (mismatch).

Most programming languages offer simple structures to build hash tables. It is easy to find efficient implementations of hash tables on the web (e.g., SpookyHash [101] or SparseHash [102]). Some tools implementing hash tables are specifically designed for routine similarity search on NGS data such as RAPSearch2 [103], which allows similarity searches in proteins and is fully compatible with Blast. Implemented in C++, the source code is freely available for download at the RAPSearch2 website [104].

The main advantage of hash tables is their speed, in particular when collisions are absent or rare. Another important advantage stands in the fact that hash tables are dynamic and can host any additional piece of information for each of its items. Unlike the FM-Index, any new item can be added to a hash table, even after its construction. In a biological context, to each k -mer can be associated its list of occurrences in a genome for instance. In comparison, the FM-Index gives only access to the occurring position(s) of each query.

The theoretical indexing and querying times (respectively $O(n)$ and $O(m)$ in average with n being the size of the bank and m the size of the query) are the same for FM-Index and hash table approaches. However, the application range are a bit different. First the hash table contains *fixed items*. For instance if items are k -mers, a hash table does not allow to query $k+1$ -mer or $k-1$ -mer. Moreover, these items must be *explicitly stored*. For a human genome, storing three billion of 31-mers (coded on a binary

alphabet) requires nearly 22 GB of memory. Indexing all 31-mers of a human genome would thus require approximately 23 GB of memory using a hash table (including the array itself). In comparison, the FM-Index is a self-indexed compressed data structure. This means that the index itself contains the original sequence, and furthermore any k -mer of any size could be queried using this data structure. Indexing a human genome using the FM-index requires less than 2 GB of memory. The BWA methods described in ref. [105] presents in details how the Burrows-Wheeler Transform (BWT) (often simply referred inaccurately as the FM-Index) can be used for performing the mapping of reads on a genome.

All the data structures presented above are designed for exact queries. However they are not made for answering questions such as “*where does $Q=TAAT$ occur with at most one error in $S=ATTTAATAAC$?*” Some attempts to make them able to deal with such requests basically need to enumerate all possibilities. For instance, the previous request would be answered by searching all queries distant by at most one error with respect to Q , representing $4^{*}(\text{size of } Q)$ queries: AAAT, CAAT, GAAT, TAAT, ACAT, etc. If more than one error should be tolerated, then the number of queries to perform explodes, making such solutions inapplicable for biological application. The search for approximated words requires additional techniques, as described in the next subsection.

3.6 Finding Approximated Words, a Matter of Seeds...

In terms of sequence approximation, two main distances are commonly used: the Hamming distance in which only substitutions are allowed and the Levenshtein distance in which insertions and deletions (called *indels*) are authorized in addition to substitutions. The Hamming distance between two words is particularly simple to compute: each couple of characters of the two words is read simultaneously and when the two characters differ, the distance is increased by one. Conversely, the Levenshtein distance (also called edit distance) is much more complex to measure. It involves a recursive organization of partial computations called *dynamic programming*. More precisely, computing the Levenshtein distance between two sequences requires a number of comparisons proportional to the product of their lengths, which becomes prohibitive when asking long queries in large banks. It is theoretically impossible to get rid of this complexity, however it is possible in practice to propose some techniques that look for an incomplete result. The goal becomes to find most of the solutions at the price of possibly missing some of them (“false negatives”), using techniques called *heuristics*. One of the most famous heuristics used in computation biology is BLAST [17], which is designed to find approximate occurrences (hits) of a query in databanks. The BLAST approach is representative of seed-based algorithms. In short, it uses the fact that two words similar enough exactly share at least some small sub-word, called the seed. For instance,

consider *TACACCCTAG* and *TCACCGCTTG*. These two words are similar and they share the seed *CACC*:

TACACC-CTAG

| | | | | | | . |

T-*CACCGCTTG*

The seed-based algorithms use this simple idea in order to speed-up their computations. Given a query and a database, they first search for occurrences of sub-words of the query in the database. This first step is performed extremely fast, using indexing techniques presented above. The positions where the shared seeds occur in the query and in the database make it possible to limit the search space in which the query may have a hit. A dynamic programming computation is performed in a second step that searches this limited space. It is always useful to keep two warnings in mind when using heuristics: they may miss some solutions and default parameters are not always the best solution (for instance, the standard seed of size 11 is insufficient for finding weak homologies between ancient interspersed repeats and it is recommended to use 7 base seeds instead).

In the last decade, the notion of seeds has been vastly improved. Of course, the smaller the seed, the less likely one is to miss some similarities. But this has two drawbacks: the filtering effect is not very sensitive and the computation becomes slower since there may be many more spurious hits that occur by chance. Two ideas have been developed to increase the sensitivity of a seed with a fixed number of characters, allowing *spaced seeds* and using *multiple seeds*. Multiple seeds are sets of seeds that are looked for simultaneously and used together to determine an E-value of the hits. The principle of spaced seeds is to choose noncontiguous characters to build them. For the previous example, *C-CT-G* (“-“ means a don’t care position in the text) has the same number of characters (the same *weight*) and thus the same selectivity than *CACC* but its sensitivity is likely to be better because it spans a longer region (6 instead of 4). Moreover, a software program like YASS [106] offers the possibility of introducing *subset seeds*, i.e., to define some positions where nucleic acids can only take a subset of possible values. For example, transition-constrained seeds (of weight $\frac{1}{2}$) have to belong to a same class in the query and the text, either purine or pyrimidine. Noting # a match position, - a don’t care position, and @ a transition position, the default seed of YASS, of weight 9, is #@# — ## — # — ##@#. Seeds of fixed weight can be optimized for a range of similarity between sequences and for a user-defined particular family of sequences in order to maximize the hitting probability. YASS can be used online or is available for download at ref. [107]. It can filter low complexity repeats and produce the same

output format than BLAST. The paper [108] provides an example of YASS pairwise comparisons applied to a gene family encoding proteins with pentatricopeptide repeat (PPR) motifs in the radish genome.

It is possible to derive a “spaced” suffix array from a standard suffix array that takes into account don’t care positions by applying a suitable transformation on the text [109] and the query. The seed-based algorithms enable the detection of repeats within a sequence or between sequences. Indeed, the algorithmic “engine” based on seeds is also adapted for comparing two long sequences in order to search for similar sub-sequences [local alignment [110] and Mummer [111]] and for comparing a sequence against itself in order to search for repeats inside this sequence itself as, for instance, is the case in the Repseek software [14].

3.7 Using Short Sequence Reads Instead of Contigs

The previously presented concepts are based on the use of queries which are smaller in length than those of the bank sequences. With the arrival of NGS (Next-Generation Sequencers) it is not uncommon to have to deal with unassembled data. In this context, queries and/or banks are composed of short sequences called reads of at most a few hundreds nucleotides. A set of such reads typically represents (all chromosomes of) an original genome. The read representation of a whole genome is neither adapted to human, nor to standard automatic analyses. For example, it becomes more difficult to compute the answer to the simple query: “Does this sequence occur in this set of reads?” and previous indexation methods must be adapted or, more radically, new sequences must be designed to be able to cope with such data representation.

When dealing with NGS reads, in particular while looking for repeats in a set of reads, there are several approaches that can be distinguished depending on the presence of a third-party reference genome or not. Given a set of sequenced reads, and when a *reference genome* exists, this latter can be used as a bank and reads are used as queries. For each read, a BLAST-like search is performed in order to know where it occurs on the genome. This process is known as *read mapping*. As it is applied to millions of reads, the mapping must be answered in a short time. Numerous methods, specific to read mapping, have been developed in the past few years. They are adapted to the size of the requests and to their expected error profiles and high similarity to the reference. A great deal of information can be extracted from the mapping of reads, especially concerning polymorphism (SNPs but also repeated elements). A good read mapper in the case of highly polymorphic genomes is the NextGenMap software [112]. The reference genome is indexed in a hash table. There exists a GitHub site including a wiki page where the code is available for downloading on [113]. When *no reference sequence* is available the user has two choices left: either reconstruct the sequence from reads (assembly process) and apply the reference-based approaches on this

assembled sequence, or use de novo methods that seek elements of interest directly in non-assembled reads. The de novo approaches are thus useful when no reference sequence exists and when the assembly of reads is problematic or impossible. This can be the case for highly complex genomes such as plant polyploid genomes. The polyploid nature of the genomes of most of the major species of agronomic interest represents a strong barrier to analysis of the organization and variation of repeats, either for noncoding areas or for duplicated genes. The only reasonable way to conduct a repeat study in these genomes seems to be to extract the repeat family of interest by careful primer design and PCR amplification [143]. However, the emergence of successful tools for de novo detection of elements of interest in raw unassembled reads can be seen. For instance, simpler to detect than repeats, the SNP detection can be performed de novo with recent tools like Cortex [114] or discoSnp [115, 144].

We have described in the list of homology-based methods for TE identification a software, RelocaTE, which is able to look for given reference TE in a set of next-generation sequencing (NGS) unassembled reads. Tools are now available to detect Transposable Elements directly from these reads. The general idea is that repeated elements are represented by a high number of reads and read frequency may be used together with sequence similarity to assemble and regroup them into repeat families. At least three methods have been designed for this purpose, RepARK [116], RetroSeq [43], and T-lex [44].

RetroSeq is a sophisticated method that can exploit mate pairs. Instead of the classic FASTA file, the input file of *RetroSeq* is a BAM format file. First, *RetroSeq* looks for discordant mate pairs: regions present in BAM sequences but not present in the reference sequence. These regions are identified as transposable elements by aligning them against the consensus library with the Exonerate software [117]. *T-lex* first uses RepeatMasker [26] to remove the TE present in highly repetitive regions from the list of TE insertions and detects the insertion of TE copies by comparing the two Target Site Duplications (TSD) and the termini of the TE from the NGS reads with the reference sequence. *T-lex* detects the deletion of TE copies by the deletion of termini regions in NGS reads.

4 Towards a Richer Characterization of Repeated Structures

4.1 A Gentle Introduction to the Theory of Languages

As can be observed in this chapter, most of the current practice of pattern matching looks at efficient ways to index and compare sequences. This has proved very useful and remains extremely important for the efficiency of any search algorithm. However, it proves to be insufficient as the knowledge and understanding of some functional or structural aspects of the different repeat

families increases. Analysts in molecular biology progressively shift from mere classification tasks to modeling tasks and develop complex scripts in order to fulfill their search needs. Programming scripts may become a tedious task because people need to express various hypothetical models of sequence architectures. It is widely acknowledged that they may even be hard to reproduce [118]. A first line of progress has been proposed with the birth and development of Scientific Workflow Management Systems [SWMS, [119]]. To search for complex patterns such as repeats, another approach is to clearly separate the descriptive part (the model of the studied sequence family) from the way this model is searched in genomes. This is precisely the goal of linguistic analysis and in the rest of this section we deal with the key concepts in this field. The description of patterns or languages on strings is the subject of the theory of formal languages. It is widely used for computer languages but can also be developed in the case of biological sequences. The search for pattern or models described in a language relies on the development of dedicated parsers that can accept any query model in the language.

The framework of *formal languages* introduces models of a possibly infinite set of sequences. The issue is to represent an infinite set in a finite way. A standard representation, called *grammar*, is a set of rewriting rules acting on a starting axiom. For instance, the following grammar (with axiom S_1) is able to recognize telomeric regions of eukaryotes like *A. thaliana*, known to be composed primarily of tandemly repeated blocks $5'-C(C|T)CTAAA-3'$:

$S_1 \rightarrow C S_2$	$S_2 \rightarrow C S_3$	$S_2 \rightarrow T S_3$	$S_3 \rightarrow C S_4$	$S_4 \rightarrow T S_5$
$S_5 \rightarrow A S_6$	$S_6 \rightarrow A S_7$	$S_7 \rightarrow A S_1$	$S_7 \rightarrow A$	

In such a model, the left part of a rule (the head) rewrites (\rightarrow) into its right part (the body). Note that rules use two types of symbols, non-terminal symbols that have to be rewritten using any rule with a matching head (S_i , $i \in [1,7]$ in our example) and terminal symbols that correspond to letters of the analyzed string (nucleic acids A, C, or T in our example). Any genomic sequence that can be generated by a finite application of such rules, starting from the axiom rule, is accepted as a telomeric region by the model. Conversely, it is possible to check (i.e., to parse) a given sequence by applying the rules from right to left on the sequence, and possibly collect some information from the parsing structure (a tree). For instance, the number of tandem repeats in the region will be the number of times S_1 occurs in the parsing tree.

It appears that the general form of rules has a deep impact on the expressiveness of the associated languages and the complexity of standard operations on these languages such as the membership of a sequence to a language or the intersection of two languages.

Furthermore, the categorization of rule types may be roughly achieved in very few classes. Thus, the rules in the example above are very specific: there exists a single, non-terminal symbol in the head and at most one non-terminal in the body, at the end of the body. It can be shown that this particular structure is characteristic of a well-known class of languages called *regular* languages. This class has been used in many pattern matching and bioinformatics tools [e.g., Unix grep for all types of text or ScanProsite [120] for proteins] and script languages (e.g., Perl) since it is possible to look for occurrences of any pattern in linear time (proportional to the length of the sequence). Often, regular expressions (e.g., $C/C|T/CTAAA$ in our example) are used instead of grammars since they offer a more compact representation but this is strictly equivalent and just a matter of notations. Despite their high utility, regular languages are limited for the recognition of repeats. They can only recognize (or serve to model) looping structures, e.g., fixed tandem repeats. Describing, for instance, the Terminal Inverted Repeats of DNA transposons is out of reach for this class of languages. Moreover, if the element that is repeated is unknown (looking for some unspecified tandem repeat or equivalently looking for all tandem repeats in a genome for instance), it is also impossible to represent the structure with a regular language.

For the case of *palindromic repeat structures*, of which the stem-loop structure in RNA sequences is one of the prime examples, it is necessary to accept grammar rules with a body containing any string of terminal and non-terminal symbols. The corresponding class of languages is called *context-free*. For instance, recognizing the TIRs of a DNA transposon could be described by the following grammar:

$S1 \rightarrow A S2 U$	$S1 \rightarrow C S2 G$	$S1 \rightarrow G S2 C$	$S1 \rightarrow U S2 A$
$S2 \rightarrow A S2 U$	$S2 \rightarrow U S2 A$	$S2 \rightarrow C S2 G$	$S2 \rightarrow G S2 C$
$S2 \rightarrow A S2$	$S2 \rightarrow C S2$	$S2 \rightarrow G S2$	$S2 \rightarrow U S2$
			$S2 \rightarrow \epsilon$

Context-free rules in this grammar (e.g., $S2 \rightarrow A S2 U$) serve to describe the Watson–Crick pairing of nucleic bases. Thus this logical structure on sequences may be clearly associated with a meaningful structure in space corresponding to chemical bonds. Other regular rules (e.g., $S2 \rightarrow A S2$) describe the internal sequence between TIRs without further constraint. The last rule $S2 \rightarrow \epsilon$ is a termination rule, where ϵ denotes the empty string. The programming languages are generally context-free languages (if you look at the html or xhtml code of a web page for instance, you will see the very same structure of pairing tags that are characteristic of context-free languages). This class of languages is more expressive but at some cost: recognizing if a model occurs in a sequence of size n may require in the worst case in the order of n^3 operations.

The next question is to know if this class of language is sufficient for biological modeling. The answer is clearly no. Consider for instance the description of chloroplast microsatellites. These “simple sequence repeats” that are stretches of small words (size less than 7 generally) are complex from the point of view of structure: it is not possible to decide in advance the number of copies or the size of copies. It requires more advanced grammatical rules, called *context sensitive*, where the non-terminal symbol on the left and the body of the rule may be surrounded by as many symbols as necessary (there exists a context of rule application) providing that the same symbols are on both sides. Other examples of context sensitive models in biological sequences are the pseudo-knot structures in RNA or the disulfide bridge structure in proteins and the introduction of errors in repeat is another source of complexity. The cost of models in this category may become very high but fortunately it is not necessary to use the full expressive power of context-sensitive languages. In practice, the art of linguistic analysis entails getting the right tradeoff between the flexibility of the modeling language and the efficiency of model parsing. From the user’s point of view, a number of models have to be tried, tuning them iteratively in order to get a reasonable number of hits. Moreover, since parsers can provide not only the hits but also their internal structure, it may be necessary to filter in post-treatment structural alternatives that are not relevant for the biological analysis.

4.2 Linguistic Analysis of Genomic Sequences

Once a language has been chosen for expressing models or patterns, any model can be searched in a bank of sequences using dedicated software. If the language is simple and specific to a sequence family, the query may generally be described by a string on a special alphabet and this task is referred to as pattern matching. If the language is more generic and allows the expression of more complex structures, it can take several rules to describe the language and the software is then called a *parser*.

4.3 Dedicated Pattern Matching

It is not possible here to provide an exhaustive review of the profusion of specific tools that have been made available to bioanalysts. Some are specific to a sequence family and others to a particular motif type.

We have already cited *ScanProsite* [120] as an example of a pattern matcher using motifs defined on a subset of regular languages. The current version accepts Prosite patterns, user-defined patterns in the Prosite syntax, a combination of patterns using logical operators *and*, *or* and *not*, and can use contextual annotation templates (ProRules) to detect functional characteristics. They are searched either by a query in a precomputed database or with an algorithm called *ps-scan*. A website is available on the Expasy server but for a large-scale independent analysis, it is possible to download the *ps-scan* Perl script [121]. A higher level parser has been developed for

the de novo recognition of human polymerase II promoter regions in ref. [122]. This study uses a two-level grammar. A regular grammar first allows recognition of promoter elements such as the TATA-box, the Initiator and the Downstream Promoter Element (DPE), etc. Then a context-free grammar is in charge of the recognition of a correct assembly of all these elements in a reasonable promoter. Unfortunately, although it is likely the authors use a generic context-free parser for this task, no tool is made available: we cite it here mostly for the purposes of illustration since it is characteristic of the linguistic approach.

A number of tools are dedicated to RNA sequences, in response to the increasing need for structure exploration in the complex RNA world, boosted by the recent importance of noncoding RNA studies. This is useful for checking structural features in retrotransposons. *RNAmotif* [123] is probably the most popular in this category as it combines a pattern description language and a language to tune the scoring. It has been designed for the description of patterns as a succession of content-constrained stems and loops, offering the possibility of choosing the standard Watson–Crick pairing (*A-U*, *G-C*) or any other user-defined pairing. The code is available for download at ref. [124]. The tool *Locomotif* [125] has almost the same expressiveness (slightly less) as the previous one but proposes interesting additional features. The first is that it allows the user to graphically design his pattern in an editor by composing several stems and loops annotated with information on the sequence content and size. A dedicated parser is automatically derived from the graphical representation provided. A second feature filters a single matching result by optimizing a thermodynamical model. A more recent tool in this category, *Structator* [126], is representative of this new generation of tools that first use a lexical analysis to significantly improve the parsing time in a second step. It makes use of an index data structure that is suited to the analysis of palindromic structures and is derived from those we have presented, the *affix array*.

4.4 General Purpose Pattern Matching

Some tools have been designed for the analysis of several types of sequences (DNA, RNA, proteins) with a generic expressiveness, i.e., without targeting the recognition of a particular motif family. Among these general tools, two tendencies can be observed, efficiency-oriented and expressiveness-oriented software.

One of the most advanced software solutions from the point of view of *efficiency* is *Vmatch* [68], which offers a wide variety of search facilities in very large sequences. *Vmatch* is a package maintained since 2003 by S. Kurtz and resulting from long experience in the field of indexing and pattern matching for genomic sequences (the initial version was called REPuter). *Vmatch* is free for academic research and can be obtained by downloading a license agreement form. It proposes a flexible command language

with numerous constructions offering a very broad variety of possible queries. It is based on a careful implementation of enhanced suffix arrays [127] for the computation of a sequence index that provides fast access to every substring in that sequence. If the search for a motif contains some rare substrings, this technique is particularly efficient. As in the previous version, REPuter, Vmatch goes from exact to approximated strings with a fixed number of mismatches by using a dynamic programming algorithm and proposes a graphical interface for the bioanalyst.

The software Vmatch is the core search engine used in a number of more specialized tools working on specific sequence structures (e.g., tandem-repeats or LTR retrotransposons in MASiVE). It is used in some databases to generate genomic information or to propose extended search functionalities. For instance, in MIPSPlantsDB, the curation and clustering in the mips-REdat repeat database [128] has been achieved using Vmatch: repeats are put in a same cluster if they share 98 % identity and the representative of each cluster is its longest sequence, a choice that makes it possible to remove incomplete sequences included in a cluster representative. PlantGDB proposes a server also based on Vmatch, *PatternSearch* [129], to look for short patterns with mismatches in *A. thaliana* or *O. sativa* genomes.

Another highly generic tool, although less expressive, is *Biogrep* [130], designed by MIT with the objective of quickly recognizing a large set of simple motifs (typically more than 100) in biological sequence banks, using multi-processor optimizations. Biogrep allows queries in the POSIX language, a standard format of extended regular expressions, and can look for patterns in parallel on a set of processors.

The other approach for the analysis of biological sequences is more concerned with modeling the peculiarities of biological objects in the most relevant and *expressive* way. A major contribution in this respect is the work of D. Searls who laid the foundations for research in this domain. He was the first to supervise developments allowing users to design biological grammars and to apply them for the large-scale analysis of their genomic sequences [131, 132]. One of D. Searls' key ideas is to try to find a balance between the well-founded framework of context-free languages that offer a good expressivity/efficiency trade-off, and the necessity of easily describing basic biological mechanisms such as copy that lie at the core of genome evolution. D. Searls introduced a very practical object in algebraic grammars, the *string variable*, which elegantly expresses this notion of copy (either direct or reverse). He has implemented the resulting logic formalism, called SVG—for StringVariable Grammars—in the (no longer available) GenLang tool [133]. From the point of view of expressivity on biological sequences, this makes it possible to take into account not only various forms of copy, distance, position, and size

constraints but also hierarchical aspects of genomic structures. For instance, in the case of LTR Retrotransposons, the top-level rule of the grammar could be represented by the following expression—this is given for the purposes of illustration only and does not pretend to be fully realistic:

LTRR→	DR:[2..6], «tg», (U5,R,U3):[80..750], «ca »,
	[1..100], pbs, [1..100], gag, [1 000..15 000], ppt, [1..100],
	«tg», (U5:80 %, R:90 %, U3:80 %), «ca», DR .

In this expression, DR, U5, R, and U3 are string variables. Its meaning is “the sequence is surrounded by two exact copies of a direct repeat (DR) of size between 2 and 6”. The LTR start with nucleotides “tg”, end with nucleotides “ca” and are made up of three parts named A, R and B with a total length between 80 and 750. The right LTR is an approximate copy of the left one. The central part (R) is the most preserved—because of the hybridization between both Rs during duplication— with a 90 % minimum identity level whereas U3 and U5 only need to have 80 % level identity. The central part of the sequence must contain at constrained distances a primer binding site (pbs), a group-specific antigen (gag), and polypurine tract (ppt), which are described by other grammatical rules.

GenLang is no longer available but *PatSearch* [134] is a restricted tool belonging to this family. It is based on the C program *scan_for_matches*, mainly written by R. Overbeek and which is downloadable from ref. [135]. It allows to describe approximated strings (including IUB codes for ambiguous nucleotides and mismatch/indel errors), gaps and length constraints, stem/loops structures and alternative patterns. Insofar as regards repeats, they can be described by a statement

```
nmax>repeat(patternident=pattern) dmin..dmax>nmin,
```

where *nmin* and *nmax* are integers fixing a range for the number of patterns, *dmin* and *dmax* fix a range for the edit distance between repeat units, and *patternident* and *pattern* are a string variable and a pattern constraining the content of this variable respectively. The keyword *frepeat* has to be used instead of *repeat* in case of exact repeat. In addition, *PatSearch* provides an assessment of the motif significance from a simulation experiment using Markov chains (estimating the number of instances that can be expected randomly).

Logol [136] is a highly descriptive language dedicated to the modeling of biological sequences and also derived from SVG. Starting from the sound basis of SVG grammars, the Logol language proposes several extensions—most notably by adopting a constraint approach—with the aim of allowing the expression of realistic biological motifs. Models use constrained string variables

(supporting overlaps, substitution, and distance errors) that can be subject to various transformations (e.g., inverse complement), gaps, and repetitions of a pattern along the sequence, negation, and alternatives to define different possibilities. As in every formal grammar components can be grouped with a view to obtaining a high-level representation of a subset of components.

Repeats may be described either with string variables or with special repeat constructs.

For example, the following model with *string variables* *I1* and *I2* can be used to look for three instances of the same string successively deriving from each other (e.g., *I1=aaaaa*, *I2=aaaca* and *I3=agaca*):

$$X1:\#[5,8]_{-I1}, *:\#[1,7], \quad ?I1:\{I2:\{I1\}, *:\#[1,7], ?I2:\{I1\}$$

The second pattern, *?I1: {I2: {I1}}*, reads as follows: the expected string must be similar to the previous *I1* string (*aaaaa* in our example), apart from 1 mismatch (*{I1}*). The matched string (*aaaca*) is saved in *I2* (*{I2}*) for further use in the last pattern (*{?I2}*). This individualization of instances means it is possible to adjust fine notions of sequence evolution.

The following example shows how palindromic repeats for the recognition of stem-loops whose stem length varies between 5 and 11 and loop size between 1 and 9 are represented. In this example, the Watson–Crick pairing is not required to be perfect: up to 2 substitutions and 1 indel are allowed.

$$STEM1:\#[5,11]_{-IS1}, \quad . *:\#[1,9], \quad -"wc" \\ ?IS1 : \{ \$[0,2], \$\$[0,1] \}$$

The content of *STEM1* (first strand of the stem) is saved in *IS1*, (*_{IS1}*). The second stem strand is then defined as the exact reverse complement of the previous content (that is *- "wc" ?IS1*), except for 2 mismatches and 1 indel.

The special constructor *repeat*, as in PatSearch, manages the characteristics of a series of occurrences. Its standard format is:

$$repeat(\langle entity \rangle, \langle distance \rangle) + \langle occurrence\ number \rangle.$$

For instance, *repeat("acgt", [0,3]) + [7,38]* states that substring *acgt* is repeated from 7 to 38 times, using a spacing of at most three characters between two repeats.

Logol is available as a web application and for download on [137]. It includes a graphical editor. In the case of spacers in a model, Logol calls on an external program using indexing sequence techniques to directly look for positions of subsequent words. Two possibilities are offered by Logol to perform indexing, either Vmatch or Cassiopee, a Ruby tool specifically developed for Logol and which is generally not as efficient as Vmatch but enables installation independently of Vmatch.

References

1. Barghini E et al (2014) The peculiar landscape of repetitive sequences in the olive (*Olea europaea* L.) genome. *Genome Biol Evol* 6:776–791. doi:[10.1093/gbe/evu058](https://doi.org/10.1093/gbe/evu058)
2. Novák P et al (2013) RepeatExplorer: a Galaxy-based web server for genome-wide characterization of eukaryotic repetitive elements from next-generation sequence reads. *Bioinformatics* 29:792–793. doi:[10.1093/bioinformatics/btt054](https://doi.org/10.1093/bioinformatics/btt054)
3. Lim KG et al (2013) Review of tandem repeat search tools: a systematic approach to evaluating algorithmic performance. *Brief Bioinform* 14:67–81. doi:[10.1093/bib/bbs023](https://doi.org/10.1093/bib/bbs023)
4. Nakamura K et al (2011) Sequence-specific error profile of Illumina sequencers. *Nucleic Acids Res* 39:e90. doi:[10.1093/nar/gkr344](https://doi.org/10.1093/nar/gkr344)
5. Luo C et al (2012) Direct comparisons of Illumina vs. Roche 454 sequencing technologies on the same microbial community DNA sample. *PLoS One* 7:e30087. doi:[10.1371/journal.pone.0030087](https://doi.org/10.1371/journal.pone.0030087)
6. Jurka J et al (2005) Repbase Update, a database of eukaryotic repetitive elements. *Cytogenet Genome Res* 110:462–467
7. Bergman CM, Quesneville H (2007) Discovering and detecting transposable elements in genome sequences. *Brief Bioinform* 8(6):382–392
8. Kurtz S et al (2008) A new method to compute K-mer frequencies and its application to annotate large repetitive plant genomes. *BMC Genomics* 9:517
9. Kurtz S et al (2001) REPuter: the manifold applications of repeat analysis on a genomic scale. *Nucleic Acids Res* 29(22):4633–4642
10. Volfovsky N, Haas BJ, Salzberg SL (2001) A clustering method for repeat analysis in DNA sequences. *Genome Biol* 2(8):RESEARCH0027
11. Morgulis A et al (2006) WindowMasker: window-based masker for sequenced genomes. *Bioinformatics* 22(2):134–141
12. Marçais G, Kingsford C (2011) A fast lock-free approach for efficient parallel counting of occurrences of k-mers. *Bioinformatics* 27:764–770
13. Gu W et al (2008) Identification of repeat structure in large genomes using repeat probability clouds. *Anal Biochem* 380(1):77–83
14. Achaz G et al (2007) Repseek, a tool to retrieve approximate repeats from large DNA sequences. *Bioinformatics* 23(1):119–121
15. Kurtz S, Myers G (1997) Estimating the probability of approximate matches. In *Proceedings of 8th symposium on combinatorial pattern matching*, Aarhus, Denmark, June/July 1997. Lecture notes in computer science, vol 1264. Springer, pp 52–64
16. Zerbino DR, Birney E (2008) Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome Res* 18:821–829
17. Altschul SF et al (1990) Basic local alignment search tool. *J Mol Biol* 215(3):403–410
18. Edgar RC, Myers EW (2005) PILER: identification and classification of genomic repeats. *BMC Bioinformatics* 9:18
19. Bao Z, Eddy SR (2002) Automated de novo identification of repeat sequence families in sequenced genomes. *Genome Res* 12(8):1269–1276
20. DeBarry J, Liu R, Bennetzen J (2008) Discovery and assembly of repeat family pseudomolecules from sparse genomic sequence data using the Assisted Automated Assembler of Repeat Families (AAARF) algorithm. *BMC Bioinformatics* 9(1):235. doi:[10.1186/1471-2105-9-235](https://doi.org/10.1186/1471-2105-9-235)
21. Johnson M et al (2008) NCBI BLAST: a better web interface. *Nucleic Acids Res* 36:W5–W9
22. Advanced Biocomputing, LLC (2009) AB-BLAST [En ligne]. <http://blast.advbio-comp.com/>
23. Schäffer AA et al (2001) Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements. *Nucleic Acids Res* 29(14):2994–3005. doi:[10.1093/nar/29.14.2994](https://doi.org/10.1093/nar/29.14.2994)
24. Jurka J et al (1996) CENSOR - a program for identification and elimination of repetitive elements from DNA sequences. *Comput Chem* 20(1):119–122
25. Smit AFA, Hubley R, Green P (1996–2010) RepeatMasker Open-3.0 [En ligne]. <http://www.repeatmasker.org/>
26. Tempel S (2012) Using and understanding RepeatMasker. *Methods Mol Biol* 859:29–51
27. Kennedy RC et al (2011) An automated homology-based approach for identifying transposable elements. *BMC Bioinformatics* 12:130
28. Haas BJ (2010) TransposonPSI [En ligne]. <http://transposonpsi.sf.net>
29. Robb SC et al (2013) The use of RelocaTE and unassembled short reads to produce high-resolution snapshots of transposable element generated diversity in rice. *G3* 3(6):949–957. doi:[10.1534/g3.112.005348](https://doi.org/10.1534/g3.112.005348)
30. Han Y, Burnette JM, Wessler SR (2009) TARGeT: a web-based pipeline for retrieving

- and characterizing gene and transposable element families from genomic sequences. *Nucleic Acids Res* 37(11):e78
31. Edgar RC (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res* 32(5):1792–1797
 32. Price MN, Dehal PS, Arkin AP (2009) FastTree: Computing large minimum-evolution trees with profiles instead of a distance Matrix. *Mol Biol Evol* 26:1641–1650
 33. Huang X, Madan A (1999) CAP3: a DNA sequence assembly program. *Genome Res* 9:868–877
 34. Larkin MA et al (2007) Clustal W and Clustal X version 2.0. *Bioinformatics* 23(21):2947–2948
 35. Benson G (1999) Tandem repeats finder: a program to analyze DNA sequences. *Nucleic Acids Res* 20(18):573–580
 36. Green P (1993–1996) phrap/cross_match/swat documentation [En ligne]. <http://www.phrap.org/phredphrap/general.html>.
 37. TimeLogic (2014). Decypher [En ligne]. <http://www.timelogic.com/>
 38. Smit A (2013) RMBlast [En ligne]. <http://www.repeatmasker.org/RMBlast.html>
 39. Smith JD (2010) Process_hits [En ligne]. <http://sourceforge.net/projects/processhits/files/README.txt/download>.
 40. Pereira V (2008) Automated paleontology of repetitive DNA with REANNOTATE. *BMC Genomics* 9:614. doi:10.1186/1471-2164-9-614
 41. Smith CD et al (2007) Improved repeat identification; masking in Diptera. *Gene* 389(1):1–9
 42. Bailly-Bechet M, Haudry A, Lerat E (2014) One code to find them all: a perl tool to conveniently parse RepeatMasker output files. *Mob DNA* 5:13. doi:10.1186/1759-8753-5-13
 43. Keane TM, Wong K, Adams DJ (2012) RetroSeq: transposable element discovery from next-generation sequencing data. *Bioinformatics* 29(3):389–390
 44. Fiston-Lavier AS et al (2011) T-lex: a program for fast and accurate assessment of transposable element presence using next-generation sequencing data. *Nucleic Acids Res* 39(6):e36
 45. Finn RD, Clements J, Eddy SR (2011) HMMER web server: interactive sequence similarity searching. *Nucleic Acids Res* 39:W29–W37
 46. Zhang Y, Zaki MJ (2006) SMOTIF: efficient structured pattern and profile motif search. *Algorithms Mol Biol* 1:22
 47. Morgante M et al (2005) Structured motifs search. *J Comput Biol* 12(8):1065–1082. doi:10.1089/cmb.2005.12.1065
 48. Nicolas J et al (2005) Suffix-tree analyser (STAN): looking for nucleotidic and peptidic patterns in chromosomes. *Bioinformatics* 21(24):4408–4410
 49. Estill JC, Bennetzen JL (2009) The DAWGPAWS pipeline for the annotation of genes and transposable elements in plant genomes. *Plant Methods* 5(1):8
 50. Flutre T et al (2011) Considering transposable element diversification in de novo annotation approaches. *PLoS One* 6(1):e16526
 51. Leroy P et al (2012) TriAnnot: a versatile. High performance pipeline for the automated annotation of plant genomes. *Front Plant Sci* 3:5
 52. Singh V, Mishra R (2010) RISC - Repeat Induced Sequence Changes Identifier: a comprehensive, comparative genomics-based, in silico subtractive hybridization pipeline to identify repeat induced sequence changes in closely related genomes. *BMC Bioinformatics* 11:609. doi:10.1186/1471-2164-11-609
 53. McCarthy EM, McDonald JF (2003) LTR_STRUC: a novel search and identification program for LTR retrotransposons. *Bioinformatics* 19:362–367
 54. Kalyanaraman A, Aluru S (2006) Efficient algorithms and software for detection of full-length LTR retrotransposons. *J Bioinform Comput Biol* 4(2):197–216
 55. Xu Z, Wang H (2007) LTR_FINDER: an efficient tool for the prediction of full-length LTR retrotransposons. *Nucleic Acids Res* 35:W265–W268
 56. Tu Z (2001) Eight novel families of miniature inverted repeat transposable elements in the African malaria mosquito, *Anopheles gambiae*. *PNAS* 98:1699–1704
 57. Rho M et al (2007) De novo identification of LTR retrotransposons in eukaryotic genomes. *BMC Genomics* 8:90
 58. Kronmiller BA, Wise RP (2008) TEnest: automated chronological annotation and visualization of nested plant transposable elements. *Plant Physiol* 146:45–59
 59. Price AL, Jones NC, Pevzner PA (2005) De novo identification of repeat families in large genomes. *Bioinformatics* 21(1):351–358
 60. Quesneville H, Nouaud D, Anxolabéhère D (2003) Detection of new transposable element families in *Drosophila melanogaster*. *Anopheles gambiae* genomes. *J Mol Evol* 57(1):S50–S59

61. Huang X (1994) On global sequence alignment. *Comput Appl Biosci* 10:227–235
62. Katoh K, Toh H (2008) Recent developments in the MAFFT multiple sequence alignment program. *Brief Bioinform* 9:286–298
63. Kolpakov R, Bana G, Kucherov G (2003) mreps: efficient and flexible detection of tandem repeats in DNA. *Nucleic Acids Res* 31:3672–3678
64. Pereira V (2004) Insertion bias and purifying selection of retrotransposons in the *Arabidopsis thaliana* genome. *Genome Biol* 5(10):R79
65. Ellinghaus D, Kurtz S, Willhoeft U (2008) LTRharvest, an efficient and flexible software for de novo detection of LTR retrotransposons. *BMC Bioinformatics* 9:18
66. Gremme G, Steinbiss S, Kurtz S (2013) GenomeTools: a comprehensive software library for efficient processing of structured genome annotations. *IEEE/ACM Trans Comput Biol Bioinform* 10(3):645–656
67. Darzentas N et al (2010) MASiVE: mapping and analysis of SireVirus elements in plant genome sequences. *Bioinformatics* 26(19):2452–2454
68. Kurtz S (2011) Vmatch: large scale sequence analysis software [En ligne]. <http://www.vmatch.de/vmweb.pdf>
69. Birney E, Clamp M, Durbin R (2004) Genewise and genomewise. *Genome Res* 14:988–995
70. Rho M, Tang H (2009) MGEScan-non-LTR: computational identification and classification of autonomous non-LTR retrotransposons in eukaryotic genomes. *Nucleic Acids Res* 37(21):e143
71. Lucier JF et al (2007) RTAnalyzer: a web application for finding new retrotransposons and detecting L1 retrotransposition signatures. *Nucleic Acids Res* 35:W269–W274
72. Rice P, Longden I, Bleasby A (2000) EMBOSS: the European Molecular Biology Open Software Suite. *Trends Genet* 16:276–277
73. Santiago N et al (2002) Genome-wide analysis of the Emigrant family of MITEs of *Arabidopsis thaliana*. *Mol Biol Evol* 19(12):2285–2293
74. Gordon AD (1999) Classification. Chapman & Hall, New York
75. Myers G (1998) A fast bit-vector algorithm for approximate string matching based on dynamic programming. In: Ninth combinatorial pattern matching conference, vol 1448, LNCS series. Springer, New York, pp 1–13
76. Warburton PE et al (2004) Inverted repeat structure of the human genome: the X-chromosome contains a preponderance of large, highly homologous inverted repeats that contain testes genes. *Genome Res* 14(10A):1861–1869
77. Chen Y, Zhou F, Li G, Xu Y (2009) MUST: a system for identification of miniature inverted-repeat transposable elements and applications to *Anabaena variabilis* and *Haloquadratum walsbyi*. *Gene* 436(1-2):1–7
78. Lu C et al (2012) Miniature inverted-repeat transposable elements (MITEs) have been accumulated through amplification bursts and play important roles in gene expression and species diversity in *Oryza sativa*. *Mol Biol Evol* 29(3):1005–1017. doi:10.1093/molbev/msr282
79. Han Y, Wessler SR (2010) MITE-Hunter: a program for discovering miniature inverted-repeat transposable elements from genomic sequences. *Nucleic Acids Res* 38(22):e199
80. Yang G (2013) MITE Digger, an efficient and accurate algorithm for genome wide discovery of miniature inverted repeat transposable elements. *BMC Bioinformatics* 14:186. doi:10.1186/1471-2105-14-186
81. Dongen SV (2008) Graph clustering via a discrete uncoupling process. *SIAM J Matrix Anal Appl* 30:121–141
82. Yang L, Bennetzen JL (2009) Structure-based discovery and description of plant and animal Helitrons. *Proc Natl Acad Sci U S A* 106(31):12832–12837
83. Markham N, Zuker M (2005) DINAMelt web server for nucleic acid melting prediction. *Nucleic Acids Res* 33:577–581
84. Charras C, Lecroq T (2004) Handbook of exact string matching algorithms. King's College publications, London. ISBN 0954300645
85. Weiner, P. (1973) Linear pattern matching algorithms. IEEE Computer Society Washington, DC, USA. SWAT '73 Proceedings of the 14th annual symposium on switching and automata theory, pp 1–11. doi:10.1109/SWAT.1973.13
86. Ukkonen E (1995) On-line construction of suffix trees. *Algorithmica* 14(3):249–260. doi:10.1007/BF01206331
87. Aluru S, Ko P (2006) In: Aluru S (ed) Handbook of computational molecular biology, Computer and information science series. Chapman & Hall, New York, Chapter 5 and 6
88. Välimäki N et al (2007) Compressed suffix tree--a basis for genome-scale sequence analy-

- sis. *Bioinformatics* 23(5):629–630. doi:[10.1093/bioinformatics/btl681](https://doi.org/10.1093/bioinformatics/btl681)
89. Mäkinen V (2013) Compressed Suffix Tree [En ligne]. <http://www.cs.helsinki.fi/group/suds/cst/>
 90. Manber U, Myers G (1993) Suffix arrays: a new method for on-line string searches. *SIAM J Comput* 22:935–948. doi:[10.1137/0222058](https://doi.org/10.1137/0222058)
 91. Kärkkäinen J, Sanders P, Burkhardt S (2006) Linear work suffix array construction. *J ACM* 53(6):918–936. doi:[10.1145/1217856.1217858](https://doi.org/10.1145/1217856.1217858)
 92. Nong G, Zhang S, Chan WH (2011) Two efficient algorithms for linear time suffix array construction. *IEEE Trans Comput* 60(10):1471–1484. doi:[10.1109/TC.2010.188](https://doi.org/10.1109/TC.2010.188)
 93. Shrestha AMS, Frith MC, Horton P (2014) A bioinformatician’s guide to the forefront of suffix array construction algorithms. *Brief Bioinform*. doi:[10.1093/bib/bbt081](https://doi.org/10.1093/bib/bbt081)
 94. Weiss D (2011) jsuffixarrays [En ligne]. <https://github.com/carrotsearch/jsuffixarrays>
 95. Barenbaum P et al (2013) Efficient repeat finding in sets of strings via suffix arrays. *Dis Math Theor Comput Sci* 15(2):59–70
 96. Becher V (2013) findrepset [En ligne]. <http://www.dc.uba.ar/people/profesores/becher/software/findrepset.tar.bz2>
 97. Burrows M, Wheeler DJ (1994) A block sorting lossless data compression algorithm. Digital Equipment Corporation, Palo Alto, Technical Report. 124
 98. Ferragina P, Manzini G (2000) Opportunistic data structures with applications. *FOCS '00 Proceedings of the 41st annual symposium on foundations of computer science*, pp 390–398. doi:[10.1109/SFCS.2000.892127](https://doi.org/10.1109/SFCS.2000.892127)
 99. Ferragina P, Manzini G (2001) An experimental study of an opportunistic index. *Proceedings of the twelfth annual ACM-SIAM symposium on discrete algorithms*. Society for Industrial and Applied Mathematics, Washington, DC, pp 269–278. ISBN 0-89871-490-7.
 100. Ferragina P, Navarro G (2005) Compressed indexes and their Testbeds [En ligne]. <http://pizzachili.di.unipi.it/>
 101. Jenkin B (2012) SpookyHash [En ligne]. <http://burtleburtle.net/bob/hash/spooky.html>
 102. Google (2012) Sparsehash [En ligne]. <http://code.google.com/p/sparsehash/>
 103. Zhao Y, Tang H, Ye Y (2012) RAPSearch2: a fast and memory-efficient protein similarity search tool for next-generation sequencing data. *Bioinformatics* 28(1):125–126. doi:[10.1093/bioinformatics/btr595](https://doi.org/10.1093/bioinformatics/btr595)
 104. Zhao Y, Ye Y (2014) RAPSearch2 [En ligne]. <http://omics.informatics.indiana.edu/mg/RAPSearch2/>
 105. Li H, Durbin R (2009) Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics* 25(14):1754–1760. doi:[10.1093/bioinformatics/btp324](https://doi.org/10.1093/bioinformatics/btp324)
 106. Noe L, Kucherov G (2005) YASS: enhancing the sensitivity of DNA similarity search. *Nucleic Acids Res* 33(2):W540–W543
 107. Noe L (2013) Yass [En ligne]. <http://bio-info.lifl.fr/yass/>
 108. Mora JRH et al (2010) Sequence analysis of two alleles reveals that intra- and intergenic recombination played a role in the evolution of the radish fertility restorer (Rfo). *BMC Plant Biol* 10:35
 109. Horton P, Kielbasa SM, Frith MC (2008) DisLex: a transformation for discontinuous suffix array construction. *Workshop on knowledge, language, and learning in bioinformatics, KLLBI. Pacific Rim International Conferences on Artificial Intelligence (PRICAI)*. pp 1–11
 110. Smith TF, Waterman MS (1981) Identification of common molecular subsequences. *J Mol Biol* 147(1):195–197. doi:[10.1016/0022-2836\(81\)90087-5](https://doi.org/10.1016/0022-2836(81)90087-5)
 111. Kurtz S et al (2004) Versatile and open software for comparing large genomes. *Genome Biol* 5(2):12. doi:[10.1186/gb-2004-5-2-r12](https://doi.org/10.1186/gb-2004-5-2-r12)
 112. Sedlazeck FJ, von Rescheneder P, Haeseler A (2013) NextGenMap: fast and accurate read mapping in highly polymorphic genomes. *Bioinformatics* 29(21):2790–2791. doi:[10.1093/bioinformatics/btt468](https://doi.org/10.1093/bioinformatics/btt468)
 113. Sedlazeck FJ, Rescheneder P (2014) NextGenMap [En ligne]. <http://cibiv.github.io/NextGenMap/>
 114. Iqbal Z et al (2012) De novo assembly and genotyping of variants using colored de Bruijn graphs. *Nat Genet* 44(2):226–232. doi:[10.1038/ng.1028](https://doi.org/10.1038/ng.1028)
 115. Peterlongo P (2014) discoSnp [En ligne]. <http://colibread.inria.fr/software/discosnp/>
 116. Koch P, Platzer M, Downie BR (2014) RepARK—de novo creation of repeat libraries from whole-genome NGS reads. *Nucleic Acids Res* 42(9):e80. doi:[10.1093/nar/gku210](https://doi.org/10.1093/nar/gku210)
 117. Slater GS, Birney E (2005) Automated generation of heuristics for biological sequence comparison. *BMC Bioinformatics* 6:31. doi:[10.1186/1471-2105-6-31](https://doi.org/10.1186/1471-2105-6-31)

118. Ioannidis JPA et al (2009) Replication of analysis of published microarray gene expression analyses. *Nat Genet* 41(2):149–155. doi:[10.1038/ng.295](https://doi.org/10.1038/ng.295)
119. Wolstencroft K et al (2013) The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud. *Nucleic Acids Res* 41(W1):W557–W561. doi:[10.1093/nar/gkt328](https://doi.org/10.1093/nar/gkt328)
120. de Castro E et al (2006) ScanProsite: detection of PROSITE signature matches and ProRule-associated functional and structural residues in proteins. *Nucleic Acids Res* 34(Web Server issue):W362–W365. doi:[10.1093/nar/gkl124](https://doi.org/10.1093/nar/gkl124)
121. de Castro E (2002) ps_scan [En ligne]. ftp://ftp.expasy.org/databases/prosite/ps_scan/
122. Datta S, Mukhopadhyay S (2013) A composite method based on formal grammar and DNA structural features in detecting human polymerase II. *PLoS One* 8(2):e54843. doi:[10.1371/journal.pone.0054843](https://doi.org/10.1371/journal.pone.0054843)
123. Macke T et al (2001) RNAMotif: A new RNA secondary structure definition and discovery algorithm. *Nucleic Acids Res* 29(22):4724–4735. doi:[10.1093/nar/29.22.4724](https://doi.org/10.1093/nar/29.22.4724)
124. Macke T (2010) RNAMotif [En ligne]. <http://casegroup.rutgers.edu/casegr-sh-2.5.html>
125. Reeder J, Reeder J, Giegerich R (2007) Locomotif: from graphical motif description to RNA motif search. *Bioinformatics* 23(13):392–400. doi:[10.1093/bioinformatics/btm179](https://doi.org/10.1093/bioinformatics/btm179)
126. Meyer F et al (2011) Structator: fast indexed search for RNA sequence-structure patterns. *BMC Bioinformatics* 12:214. doi:[10.1186/1471-2105-12-214](https://doi.org/10.1186/1471-2105-12-214)
127. Abouelhoda MI, Kurtz S, Ohlebusch E (2004) Replacing suffix trees with enhanced suffix arrays. *J Dis Algorithms* 2(1):53–86. doi:[10.1016/S1570-8667\(03\)00065-0](https://doi.org/10.1016/S1570-8667(03)00065-0)
128. Nussbaumer T et al (2013) MIPS PlantsDB: a database framework for comparative plant genome research. *Nucleic Acids Res* 41(Database issue):D1144–D1151
129. Brendel V (2007) Pattern Search [En ligne]. <http://www.plantgdb.org/cgi-bin/vmatch/patternsearch.pl>
130. Jensen K, Stephanopoulos G, Rigoutsos I (2002) Biogrep: a multi-threaded pattern matcher for large pattern sets. *kljensen/biogrep* GitHub [En ligne]. <https://github.com/kljensen/biogrep>
131. Searls DB (2002) The language of genes. *Nature* 420(6912):211–217
132. Searls DB (1995) String variable grammar: a logic grammar formalism for DNA sequences. *J Log Program* 24(1–2):73–102
133. Dong S, Searls DB (1994) Gene structure prediction by linguistic methods. *Genomics* 23:540–551
134. Grillo G et al (2003) PatSearch: a program for the detection of patterns and structural motifs in nucleotide sequences. *Nucleic Acids Res* 31(13):3608–3612. doi:[10.1093/nar/gkg548](https://doi.org/10.1093/nar/gkg548)
135. Overbeek R (2010) ScanForMatches [En ligne]. <http://blog.theseed.org/servers/2010/07/scan-for-matches.html>
136. Belleannée C, Sallou O, Nicolas J (2012) Expressive pattern matching with Logol. Application to the modelling of -1 ribosomal frameshift events. *JOBIM'2012, Rennes*. pp 5–14. http://jobim2012.inria.fr/jobim_actes_2012_online.pdf
137. Sallou O (2014) Logol [En ligne]. <http://logol.genouest.org>
138. Ouyang S, Buell CR (2004) The TIGR Plant Repeat Databases: a collective resource for the identification of repetitive sequences in plants. *Nucleic Acids Res* 32(Database issue):D360–D363
139. Bousios A et al (2012) MASiVEDb: the Sirevirus Plant Retrotransposon Database. *BMC Genomics* 13(158)
140. Chen J et al (2013) P-MITE: a database for plant miniature inverted-repeat transposable elements. *Nucleic Acids Res* 42(Database issue):D1176–D1181. doi:[10.1093/nar/gkt1000](https://doi.org/10.1093/nar/gkt1000)
141. Malde K et al (2006) RBR: library-less repeat detection for ESTs. *Bioinformatics* 22(18):2232–2236
142. Li R et al (2005) ReAS: recovery of ancestral sequences for transposable elements from the unassembled reads of a whole genome shotgun. *PLoS Comput Biol* 1(4):e43
143. You FM et al (2010) RJPrimers: unique transposable element insertion junction discovery and PCR primer design for marker development. *Nucleic Acids Res* 38(Suppl 2):W313–W320
144. Nakagome M et al (2014) Transposon Insertion Finder (TIF): a novel program for detection of de novo transpositions of transposable elements. *BMC Bioinformatics* 15:71. doi:[10.1186/1471-2105-15-71](https://doi.org/10.1186/1471-2105-15-71)

Analysis of RNA-Seq Data Using TopHat and Cufflinks

Sreya Ghosh and Chon-Kit Kenneth Chan

Abstract

The recent advances in high throughput RNA sequencing (RNA-Seq) have generated huge amounts of data in a very short span of time for a single sample. These data have required the parallel advancement of computing tools to organize and interpret them meaningfully in terms of biological implications, at the same time using minimum computing resources to reduce computation costs. Here we describe the method of analyzing RNA-seq data using the set of open source software programs of the Tuxedo suite: TopHat and Cufflinks. TopHat is designed to align RNA-seq reads to a reference genome, while Cufflinks assembles these mapped reads into possible transcripts and then generates a final transcriptome assembly. Cufflinks also includes Cuffdiff, which accepts the reads assembled from two or more biological conditions and analyzes their differential expression of genes and transcripts, thus aiding in the investigation of their transcriptional and post transcriptional regulation under different conditions. We also describe the use of an accessory tool called CummeRbund, which processes the output files of Cuffdiff and gives an output of publication quality plots and figures of the user's choice. We demonstrate the effectiveness of the Tuxedo suite by analyzing RNA-Seq datasets of *Arabidopsis thaliana* root subjected to two different conditions.

Key words RNA-seq, Bowtie, TopHat, Cufflinks, Cuffmerge, Cuffcompare, Cuffdiff, CummeRbund, Differential gene expression, Transcriptome assembly

1 Introduction

In the early years of transcriptome studies, microarray technologies based on nucleic acid hybridization were predominately used for gene expression analysis. However, these technologies posed a limitation in quantifying the various kinds of RNA molecules that are expressed by the genome under myriad biological conditions at different points of time. The limitation is mainly due to the reliance on extensive prior knowledge of the genome [1]. The recent advances in Next Generation Sequencing (NGS) of DNA [2] have led to a burst in RNA analysis through massively parallel cDNA sequencing that has accelerated our understanding of the dynamics of genome expression and genome structural variation [1, 3, 4]. Methods have also been devised to directly sequence RNA on a massively parallel scale without intermediate cDNA generation [5]

in order to avoid some of the pitfalls or artifacts produced during cDNA synthesis from RNA strands [6], producing reads which are bias free and more comprehensive. High throughput RNA-seq produces large volumes of data in a single experiment, which need to be assembled correctly in order to interpret them meaningfully. A large number of computation methods and tools have been developed for efficient assembly, quantification, and differential analysis of such data [7]. Of these the “Tuxedo suite” of programs—Bowtie, TopHat, and Cufflinks—along with accessory tools and programs, like CummeRbund, have gained a lot of popularity. The TopHat algorithm maps sequenced reads to a reference genome without relying on prior annotation of genes and accounts for alternative splicing of a primary transcript, while the Cufflinks algorithm estimates transcript abundances and differential expression. Analysis of large volumes of RNA-seq data with these algorithms has revealed novel isoforms and splice variants, as well as unannotated transcripts even in well studied biological models [8].

1.1 RNA-seq

The transcriptomic profile of an organism at any given time or condition gives the set of all its transcripts and their quantities present at the specific time point or condition. The transcriptome reveals a great deal about the functional aspects of the genome as well as the different kinds of biomolecules present within the cell or tissue. It is also very useful for studying the genetics behind growth, development and disease. Transcriptome studies aim at cataloging the different kinds of RNA in an organism, and quantifying the changes in expression levels of individual transcripts with time or under certain conditions [9]. Previously, there have been two approaches for sequencing RNA: (a) The hybridization based approach which involved the use of microarrays [10, 11] and (b) The sequencing approach which directly determined the cDNA sequence using Sanger Sequencing approaches. The latter approach developed as an alternative to microarray technology, because hybridization heavily relied upon known sequenced genes and genomic regions. As sequencing the whole cDNA segment to produce ESTs is expensive and has huge limitations when quantifying expressions, many tag based high throughput sequence approaches were developed for expression quantification. These included CAGE (Cap Analysis of Gene Expression) [12, 13], SAGE (Serial Analysis of Gene Expression) [14], and MPSS (Massively Parallel Signature Sequencing) [15, 16]. The advancement of high throughput NGS technology makes the RNA-seq technology feasible and affordable for transcriptome profiling. RNA-seq involves extracting the total RNA from a sample, then selecting an RNA population subset (such as poly A (+)), converting them to cDNA, and ligating the cDNA to specific adaptors at one or both ends of the fragment. These modified fragments are then sequenced from one (single end sequencing) or both ends (paired end sequencing), with or without amplification, in a high throughput NGS

sequencing machine. After sequencing, the RNA reads are mapped and quantified based on a reference genome [17]. If a reference genome is not available, de novo transcriptome assembly can be carried out [18]. The RNA-seq analysis is also good for determining spliced junctions with the resolution of a single base, determining the structural variations in the genome, and locating transcript boundaries and novel genes and transcripts [9]. However, there are weaknesses in the RNA-seq technology. For example, DNA artifacts can be synthesized during cDNA cloning and that could cause misinterpretations later on. This weakness can be remediated by carrying out RNA-seq in replicates to even out the differences or account for sample variability during analysis [6, 7]. For a comprehensive review of RNA-seq, it is recommended to review the paper by Wang et al. [9].

1.2 The Tuxedo Suite of Computational Tools

RNA-seq data are usually very large and require efficient algorithms which use minimum computing resources for their analysis.

There are three main steps in the reference-based RNA-seq analysis [7]:

1. Aligning RNA-seq reads to a reference genome or transcriptome.
2. Assembling the aligned reads into transcription units, or reconstruction of a transcriptome.
3. Performing differential expression of transcripts across different conditions or time points

Each of the three steps possesses different computation challenges [7]. Although there are several different algorithms and tools that address each of these challenges separately, in this chapter, we only illustrate the popular Tuxedo Suite which contains a comprehensive set of programs that perform reference-based RNA-seq analysis efficiently. The core programs for RNA-seq analysis in the Tuxedo suite are:

1. **Bowtie** [19], an ultrafast unspliced read aligner with high memory efficiency. It uses the Burrows Wheeler Transform [20] and FM indexing [21] to compact a reference genome into a data structure that is efficient for aligning short reads with perfect matches.
2. **TopHat** [22] uses an “exon first” approach to align reads. It uses Bowtie to map the short contiguous unspliced reads to a reference genome. The unmapped reads are then split into shorter fragments and then are aligned independently to identify splice junctions between exons.
3. **Cufflinks** [23] uses the alignment file from TopHat, or other aligners which allow spliced alignments, to assemble and reconstruct the transcriptome. It assembles the overlapping “bundles” of aligned reads into transcripts using a probabilistic

approach, then merges multiple conditions and estimates the transcript abundances using Cuffmerge and Cuffcompare respectively (*see Note 9*). It can also use **Cuffdiff** to calculate the differential gene expression.

4. Finally, an accessory tool, **CummeRbund** [24, 23], renders the Cuffdiff output into visual representations like bar, scatter and volcano plots.

The Tuxedo suite programs are all run in the command line. Alternatively, the web-based platform, Galaxy [25], provides a better user friendly interface to run the Tuxedo suite programs but the free public Galaxy server is limited to analysis of smaller datasets.

1.3 Other Tools and Comparison with Tuxedo

The Tuxedo suite provides a comprehensive workflow for RNA-seq analysis. There are a number of alternative packages and tools that can be used in conjunction with or as an alternative to any of the tools in the Tuxedo suite. For example there are many other unspliced read aligners like SHRiMP [26] and Stampy [27] that follow the seed method and Smith–Waterman extension to align unspliced reads to the genome with high accuracy, and spliced aligners like MapSplice [28] and GSNAP [29]. There are also transcriptome assemblers like Scripture [30] (which is similar to Cufflinks but uses a segmentation approach) and Velvet/Oases [31] (where Velvet assembles the reads into transcripts then Oases builds the transcript isoforms). Finally, transcript abundance can be calculated by alternative tools like ERANGE [17], and NEUMA [32] and differential expression using DEGseq [33] and DESeq [34]. Some of these tools work with minimal assumption about the library construction protocol but most RNA-seq analysis tools require paired-end reads. A good review of RNA-seq tools along with explanations of the algorithms can be found in Garber et al. [7].

The Tuxedo suite serves as a comprehensive transcriptome analysis solution and the algorithms are robust and accurate. They also have minimum computing requirements in terms of processor cores and memory compared to other programs. They have gained popularity and have been used in a number of high-profile transcriptome studies [35].

2 Materials

TopHat and Cufflinks are run using Linux command line and the CummeRbund program is run in an R interactive shell. In the following discussion, commands that were run in a Linux shell are shown with a '\$' prefix whereas those run in an R interactive shell are shown with a '>' prefix.

The software used require, for best results, a 64-bit version of the operating system, with at least 4 GB of RAM.

Before beginning the example run, it is recommended to create a working directory. Here we have used a directory called *exp_run*. Additionally we have downloaded all the tools and data in a *Downloads_tc* folder. All installation is carried out by creation of a *bin* directory in the user's home directory, which is added to the PATH environment variable. The pre-compiled executable binaries are added to this *bin* directory.

2.1 RNA-seq Data

```
$ mkdir exp_run
$ mkdir Downloads_tc
  $ mkdir $HOME/bin
  $ export PATH=$HOME/bin:$PATH
```

For the purpose of this demonstration, we have chosen an experiment on *Arabidopsis thaliana*. The raw RNA-seq data were obtained from the Sequence Read Archive (SRA) under the National Center for Biotechnology Information (NCBI) (<http://www.ncbi.nlm.nih.gov/sra>). The data are in the experiment with the series accession number GSE44062 in the Genome Expression Omnibus of the NCBI [36]. We have selected the first four sample runs (SRR671946, SRR671947, SRR671948, and SRR671949) for the purpose of this demonstration (*see Note 1*).

Thus, in this demonstration, we show the differential gene expression in *Arabidopsis thaliana* root subjected to two different conditions—one is treated with KNO₃, while the other with KCl as a control. There are two biological replicates for each condition (*see Note 2*).

The data are downloaded in the SRA format and converted to FASTQ using the SRA Toolkit (version 2.3.4-2). The precompiled toolkit package is downloaded, and the locations of executables are added to the PATH or the bin folder of the current user. Then the following command is executed to convert the zipped SRA file to FASTQ:

```
$ fastq-dump --split-3 --gzip --outdir ./ --define-qual '+' ./SRR000000.sra
```

Here, SRR000000 represents the file name.

These FASTQ data files are stored in the working directory, with their extensions changed from *.fq* to *.fastq*

```
$ cp SRR000000.fq $HOME/exp_run/SRR000000.fastq
```

2.2 Reference Genome

Bowtie and TopHat need a sequenced reference genome in order to align the raw RNA-seq reads. This genome has to be indexed before it is used. The Bowtie website has many pre-built indexes to be used for this purpose. Alternatively, Illumina's iGenome project provides Bowtie 1 and 2 indexes pre-built for a number of model organisms (<http://tophat.cbcb.umd.edu/igenomes.shtml>). In this demonstration, we shall use the *Arabidopsis thaliana* TAIR 10

package assembled by the iGenome project with sequences downloaded from Ensembl. Users have the freedom of building their own Bowtie indexes for organisms whose pre-built indexes are not available. One can do this using the build command from Bowtie.

First, you have to obtain the complete sequence file(s), then move it (or them) to the Bowtie2 unpacked directory, and type out the following command:

```
$ bowtie2-build [options]<comma_separated_list_of_FASTA_files><user's_choice_of_base_name_of_bowtie2_index>
```

The output contains a set of new files with the user specified base name.

The iGenome package for Arabidopsis (Ensembl, TAIR 10) downloaded, when unpacked, appears as *Arabidopsis_thaliana*. Within it, there is an *Ensembl* directory and then a *TAIR10* directory. In the *TAIR10* directory there are three directories: *Annotation*, *GenomeStudio*, and *Sequence*. Of these the ones relevant to this analysis are *Annotation* and *Sequence*. The necessary files are copied out into the working directory:

```
$ cp Annotation/Genes/genes.gtf path-to-exp_run
$ cp Sequence/Bowtie2Index/genome.* path-to-exp_run
```

2.3 Software and Tools

2.3.1 Bowtie Software

This demonstration utilizes Bowtie2-2.1.0 which can be downloaded from the Bowtie website (<http://sourceforge.net/projects/bowtie-bio/files/bowtie2/2.1.0/>). Download the binaries for Linux 64-bit system to the *Downloads_tc* folder as a tarball and unpack them.

```
$ unzip bowtie2-2.1.0-linux-x86_64.zip
```

From within the unpacked bowtie2 directory, copy out the executable binaries into the bin folder:

```
$ cd bowtie2-2.1.0
$ cp * path-to-USER/bin
```

2.3.2 TopHat Software

Download the Linux 64-bit binary version of TopHat-2.0.10 from the TopHat website (<http://tophat.cbcb.umd.edu/index.shtml>) to the *Downloads_tc* folder as a tarball, and unpack them.

```
$ tar -xvzf tophat-2.0.10.Linux_x86_64.tar.gz
```

From within the unpacked TopHat directory, copy out the executable binaries into the bin folder:

```
$ cd tophat-2.0.10.Linux_x86_64
$ cp * path-to- USER/bin
```

2.3.3 SAM Tools

For the SAM [37] (Sequence Alignment Map) tools, download version 0.1.19 from <http://sourceforge.net/projects/samtools/files/> to the *Downloads_tc* folder as a tarball, and unpack them.

```
$ tar -xvjf samtools-0.1.19.tar.bz2
```

The SAM tools are installed by first changing to the unpacked directory and then using the make command to compile the source code. To install SAM tools, one needs to have zlib and htlib

installed. The resultant ‘samtools’ binary is copied into the bin folder added to the PATH.

```
$ make
$ cp samtools $HOME/bin
```

2.3.4 Cufflinks Software

For the demonstration analysis, we have used Cufflinks-2.1.1, which can be downloaded from the Cufflinks website (<http://cufflinks.cbc.umd.edu/index.html>). Download the Linux 64-bit binary version to the *Downloads_tc* folder as a tarball and unpack them (*see Note 3*).

```
$ tar -xvzf cufflinks-2.1.1.Linux_x86_64.tar.gz
```

From within the unpacked Cufflinks directory, copy the executable binaries into the bin folder:

```
$ cd cufflinks-2.1.1.Linux_x86_64
$ cp * $HOME/bin
```

2.3.5 CummeRbund Software

The CummeRbund is in R script. To download it, start an R session using the following command:

```
$ R
```

Following which, the documentation for R will appear. Thereafter, type the following commands to install bioconductor and then CummeRbund: (*see Note 4*)

```
>source('http://www.bioconductor.org/biocLite.R')
>biocLite('cummeRbund')
```

3 Methods and Results

3.1 Running TopHat

Type the following commands to map the RNA-seq reads to the genome (*see Note 5*).

```
$ tophat -p 2 -i 20 -I 5000 -G genes.gtf -o SRR671946_
tophatOut genome SRR671946.fastq
```

```
$ tophat -p 2 -i 20 -I 5000 -G genes.gtf -o SRR671947_
tophatOut genome SRR671947.fastq
```

```
$ tophat -p 2 -i 20 -I 5000 -G genes.gtf -o SRR671948_
tophatOut genome SRR671948.fastq
```

```
$ tophat -p 2 -i 20 -I 5000 -G genes.gtf -o SRR671949_
tophatOut genome SRR671949.fastq
```

Here, ‘SRR000000_tophatOut’ represents the output directory for each run. If no output directory is specified by the *-o* option, TopHat automatically creates a directory called *tophat_out* in the working director, and stores the output files in it.

The successful run creates a directory with the name specified by the user, containing the following files: *accepted_hits.bam*, *align_summary.txt*, *deletions.bed*, *insertions.bed*, *junctions.bed*, *prep_reads.info*, *unmapped.bam* and a directory with the name *logs*. The *accepted_hits.bam* is the main result file containing the mapped results in binary format.

3.2 Running Cufflinks to Assemble Transcripts

Using the TopHat alignment files, we can directly run Cufflinks with the following commands to assemble the transcripts (*see Note 6*).

```
$ cufflinks -p 2 -o SRR671946_cufflinksout SRR671946_tophatOut/accepted_hits.bam
```

```
$ cufflinks -p 2 -o SRR671947_cufflinksout SRR671947_tophatOut/accepted_hits.bam
```

```
$ cufflinks -p 2 -o SRR671948_cufflinksout SRR671948_tophatOut/accepted_hits.bam
```

```
$ cufflinks -p 2 -o SRR671949_cufflinksout SRR671949_tophatOut/accepted_hits.bam
```

For each run, the designated output directory will contain the following files: genes.fpk_tracking, isoforms.fpk_tracking, skipped.gtf, transcripts.gtf. The assembled transcripts are contained in transcripts.gtf.

3.3 Running Cuffmerge to Merge the Transcripts to a Comprehensive Transcriptome

Create a file called assembled_tc.txt that lists out the file path of the assembly file (transcripts.gtf) for each sample. This can be simply done by any text editor, for example nano (*see Note 7*).

```
$ nano
```

Once the text editor is opened, put in the file paths of the four transcript assemblies:

```
./SRR671946_cufflinksout/transcripts.gtf
```

```
./SRR671947_cufflinksout/transcripts.gtf
```

```
./SRR671948_cufflinksout/transcripts.gtf
```

```
./SRR671949_cufflinksout/transcripts.gtf
```

The file is saved using the name ‘assembled_tc.txt’

The assembled transcripts are then used to run Cuffmerge using the following command:

```
$ cuffmerge -g genes.gtf -s genome.fa -p 2 assembled_tc.txt
```

The successful run creates a *merged_asm* directory, which contains a *logs* directory and a file containing the information of the merged transcripts called merged.gtf.

3.4 Running Cuffdiff to Check for Differential Expression Under Different Conditions

Type the following command in a single line (*see Notes 8 and 9*).

```
$ cuffdiff -o diff_result -b genome.fa -p 2 -L Root_Kcl_control,Root_KNO3_treatment -u merged_asm/merged.gtf ./SRR671946_tophatOut/accepted_hits.bam,./SRR671947_tophatOut/accepted_hits.bam,./SRR671948_tophatOut/accepted_hits.bam,./SRR671949_tophatOut/accepted_hits.bam
```

The successful run creates a directory *diff_result* in the working directory. The directory contains a number of different files and databases, listed as follows:

bias_params.info	cds.diff
cds.count_tracking	cds_exp.diff
cds.fpk_tracking	promoters.diff
cds.read_group_tracking	read_groups.info

gene_exp.diff	run.info
genes.count_tracking	splicing.diff
genes.fpk_tracking	tss_group_exp.diff
genes.read_group_tracking	tss_groups.count_tracking
isoform_exp.diff	tss_groups.fpk_tracking
isoforms.count_tracking	tss_groups.read_group_tracking
isoforms.fpk_tracking	var_model.info
isoforms. read_group_tracking	

The fpkm tracking files give FPKM counts of primary transcripts (tss_groups.fpk_tracking), genes (genes.fpk_tracking), coding sequences (cds.fpk_tracking), and transcripts (isoforms.fpk_tracking).

The count tracking files give the number of fragments for each gene (genes.count_tracking), transcript (isoforms.count_tracking), primary transcript (tss_groups.count_tracking) and coding sequence (cds.count_tracking).

The read group tracking files contain information on the counts of genes, transcripts and primary transcripts, grouped by replicates.

The diff files ending with ‘exp.diff’ contain information on the differential expression tests performed on the genes (gene_exp.diff), primary transcripts (tss_group_exp.diff), transcripts (isoform_exp.diff), and coding sequences (cds_exp.diff).

The promoters.diff file lists out the genes producing two or more primary transcripts, as well as the differential promoter usage of these genes across the samples, in a tabulated format.

The splicing.diff file contains information on primary transcripts that have two or more isoforms, and how they are differently spliced across the samples.

The cds.diff file provides information on the genes that are multi-protein and the extent of their differential coding sequence output between the samples.

Files ending with ‘.info’ contain information mostly on the options provided to and parameter values used by Cuffdiff for expression quantification and assessment of differential expression.

3.5 Results

The original paper from where we have sourced the data [36], used DESeq to analyze the RNA-seq data and highlighted the genes which were differentially expressed across the two conditions. In this section, we show an overall view of the differential gene expression between the two conditions using CummeRbund. Then we use the genes BT1 and ATPPC3 as an example to show

that they are differentially expressed, which is consistent with the findings of the paper [36]. We also use the AT2G33550 gene as an example to show the differential expression of its isoforms across the two conditions, and list out some genes which are similar to the gene BT1. The section also contains some other useful commands for other visualizations like generating scatter and volcano plots.

3.5.1 Running CummeRbund to Generate Visual Graphs and Publishable Figures

Start an R session in the working directory and load the cummeRbund module using the following command:

```
>library('cummeRbund')
```

The readCufflinks function in cummeRbund requires the Cuffdiff output folder, *diff_result*, as an input argument. The following will store the results in the 'cuffdata' object (*see Note 10*).

```
>cuffdata <- readCufflinks('diff_result')
```

```
>cuffdata
```

The last command prints out the Cuffdiff result summary as follows:

```
CuffSet instance with:
```

```
2 samples
```

```
33318 genes
```

```
42109 isoforms
```

```
34957 TSS
```

```
32921 CDS
```

```
33318 promoters
```

```
34957 splicing
```

```
27174 relCDS
```

Plotting Distribution of Expression Levels for Each Sample

To obtain a density plot showing the expression levels for each sample, use the csDensity function:

```
> csDensity(genes(cuffdata))
```

This gives the output of a density plot as an image file (*see Note 11*) (*see Fig. 1*).

Examining Differentially Expressed Genes Across the Conditions Using a Volcano Plot

To obtain a volcano plot showing the differentially expressed genes across the two samples type the following command:

```
>csVolcano(genes(cuffdata), 'Root_Kcl_control', 'Root_KNO3_treatment')
```

(*See Fig. 2*)

Comparing the Expression of Genes in Each Condition Using a Scatter Plot

To obtain a scatter plot comparing the gene across the two samples type the following command.

```
>csScatter(genes(cuffdata), 'Root_Kcl_control', 'Root_KNO3_treatment')
```

(*See Fig. 3*)

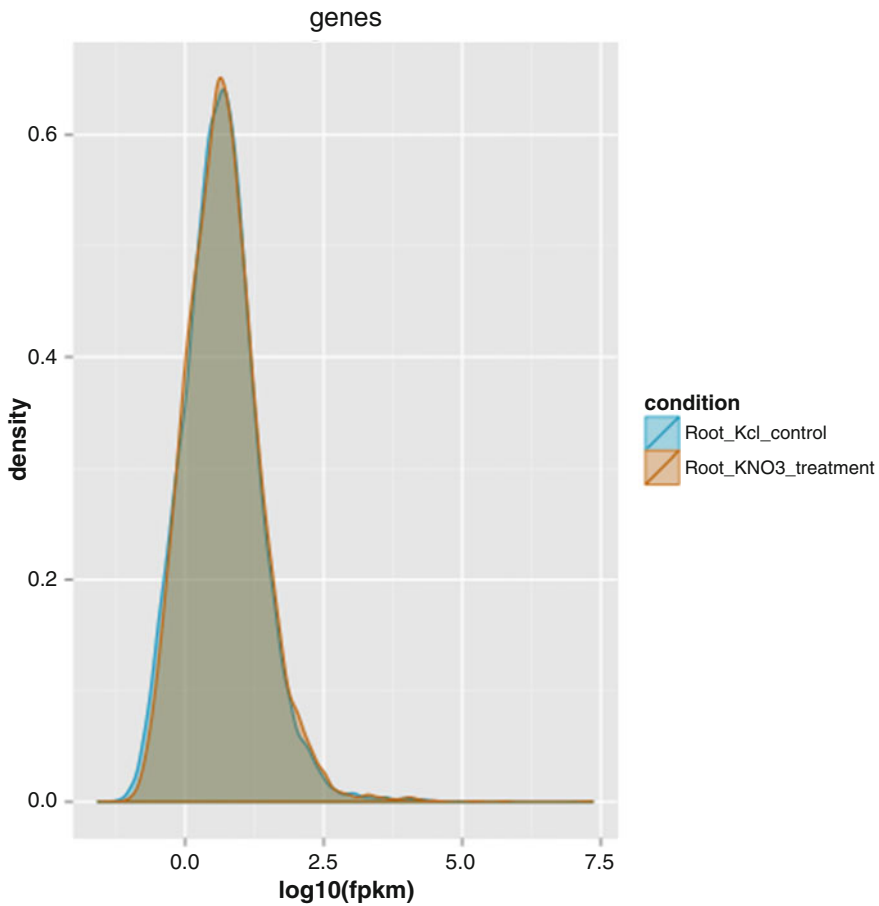


Fig. 1 Density plot of the genes of *Arabidopsis thaliana* root under treated (2 h of 5 mM KNO₃) and control (2 h of 5 mM KCl) conditions

Determining the Number
of Differentially
Expressed Genes

To find the number of differentially expressed genes, type the following command

```
> gene_diff_data <- diffData(genes(cuffdata))
> sig_gene_data <- subset(gene_diff_data, (significant == 'yes'))
> nrow(sig_gene_data)
[1] 196
```

Recording the Differentially
Expressed Genes' Details

To print a table displaying the details of all the differentially expressed genes, type out the following command.

```
> gene_diff_data <- diffData(genes(cuffdata))
> sig_gene_data <- subset(gene_diff_data, (significant == 'yes'))
> nrow(sig_gene_data)
```

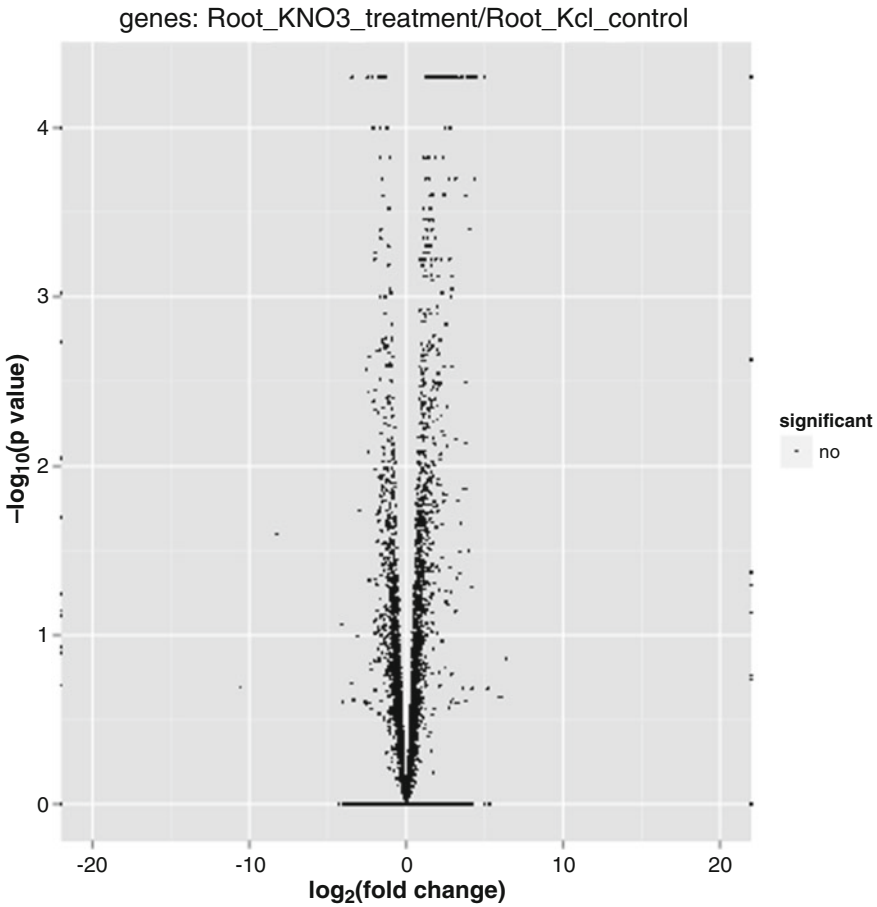


Fig. 2 A volcano plot indicating the presence of differentially expressed genes between the two samples. The plot shows that some genes show a huge change in expression when *Arabidopsis thaliana* root is treated with KNO₃

```
> write.table(sig_gene_data, 'diff_genes.txt', sep = '/t', row.names = F, col.names = T, quote = F)
> sig_gene_data
```

The last command prints out a table containing the details of all the differentially expressed genes.

3.5.2 Comparing the Expression Levels of Select Genes Using Bar Plots

Compare the gene expression levels of the BT1 gene under both conditions using the following command (*see* Fig. 4).

```
> gene_int <- getGene(cuffdata, 'BT1')
> expressionBarplot(gene_int)
```

Compare the gene expression levels of the ATPPC3 gene under both conditions using the following command (*see* Fig. 5).

```
> gene_int2 <- getGene(cuffdata, 'ATPPC3')
> expressionBarplot(gene_int2)
```

The observations obtained are consistent with the findings in the paper [36], where the authors have used DESeq to analyze the relative expression levels of the differentially expressed genes.

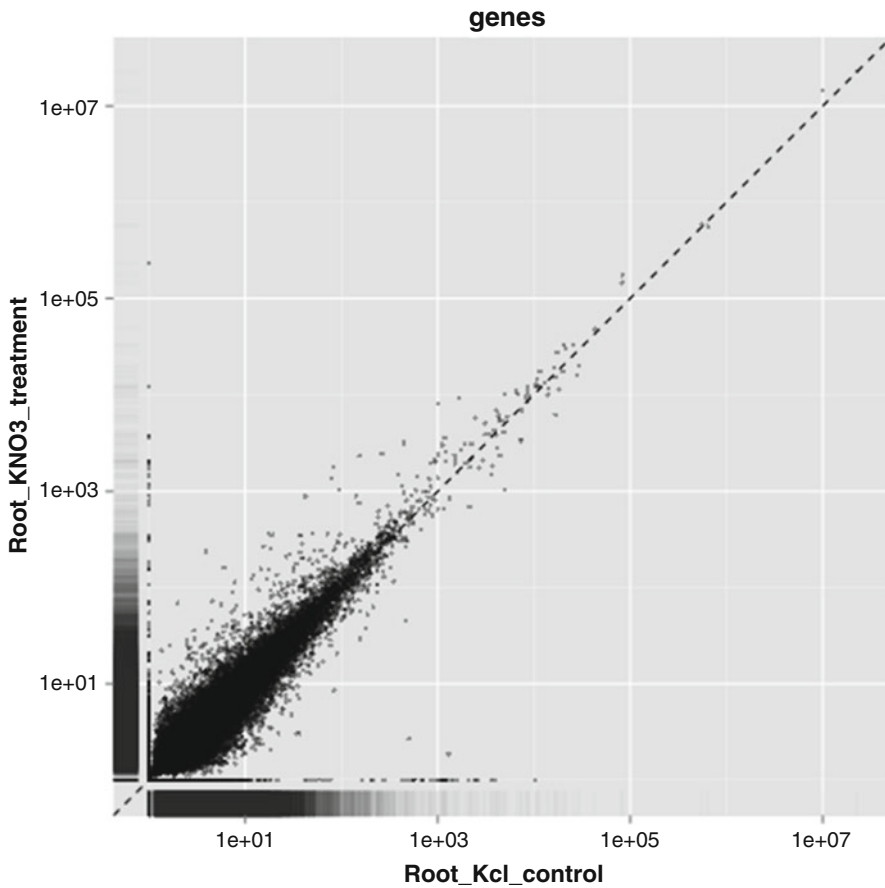


Fig. 3 A scatter plot showing the expression of genes under the two experimental conditions, with the x-axis representing the gene expression values for the control condition and the y-axis representing the gene expression values for the treated condition. Each point thus represents the expression of a gene under both conditions. From the plot, it is evident that that gene expression of some of the genes is increased in the treated condition

3.5.3 Comparing the Expression Levels of the Isoforms of Select Genes Using Bar Plots

Compare the expression levels of the isoforms of the AT2G33550 gene using the following command (*see* Fig. 6a, b).

```
> gene_int3 <- getGene(cuffdata, 'AT2G33550')
> expressionBarplot(isoforms(gene_int4))
```

It is interesting to note how the isoform expression levels roughly add up to give the total gene expression level for a given condition.

3.5.4 Finding Genes Similar to a Given Gene

To find four genes similar to the gene BT1, type out the following command (*see* Fig. 7).

```
> sim_gene <- findSimilar(cuffdata, "BT1", n=4)
> sim_gene.expression <- expressionPlot(sim_gene, log-
Mode=T, showErrorBars=F)
> sim_gene.expression
```

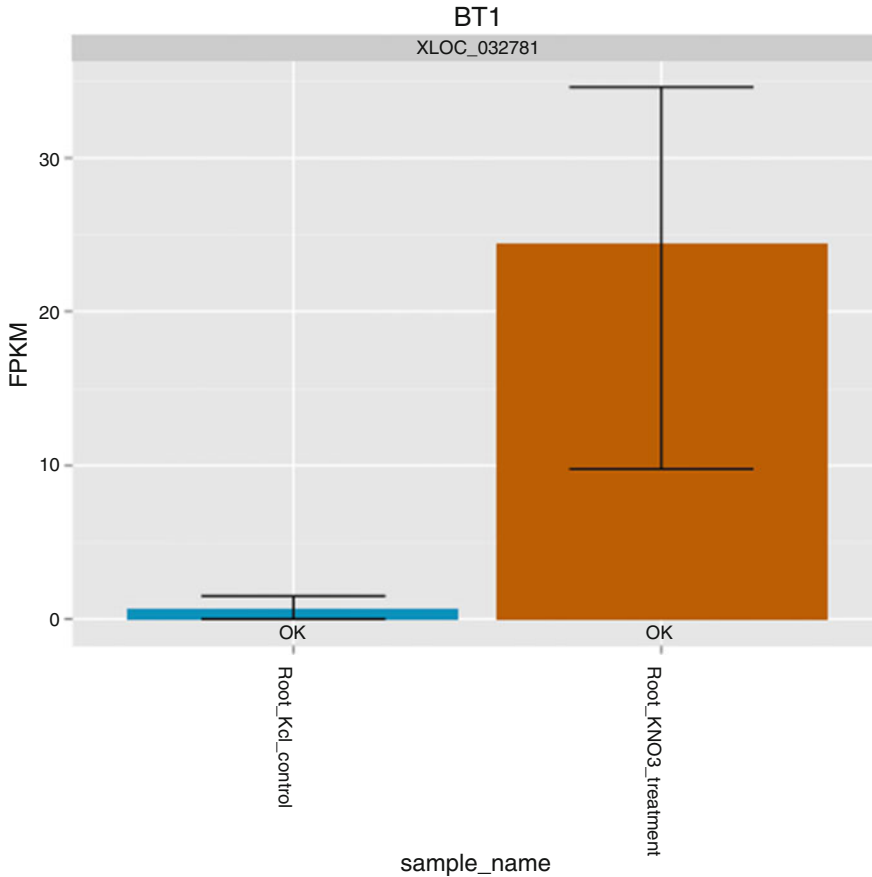


Fig. 4 A bar plot showing the expression of the gene *BT1* under treated (*orange*) and untreated (*blue*) conditions. The FPKM value shown on the *y*-axis is directly proportional to the level of gene expression, so the fold change can be estimated from the graph

4 Notes

1. An easy way to locate the data would be to go to the NCBI website (<http://www.ncbi.nlm.nih.gov>) and to search for the accession number “GSE44062” in the GEO Dataset database. On the search results that show up, click on the result that says “Analysis of the nitrate-responsive transcriptome of the Arabidopsis root”. In the sample section, click on the desired sample, and the sample page will appear with the ftp download link at the bottom of the page. The detailed information about each sample is available in the SRA run archive (http://www.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=run_browser) and can be accessed by typing in the run accession for the sample (‘SRR000000’), and then referring to the metadata tab.
2. Although TopHat and Cufflinks make only a few assumptions about the protocol for library preparation for obtaining RNA-

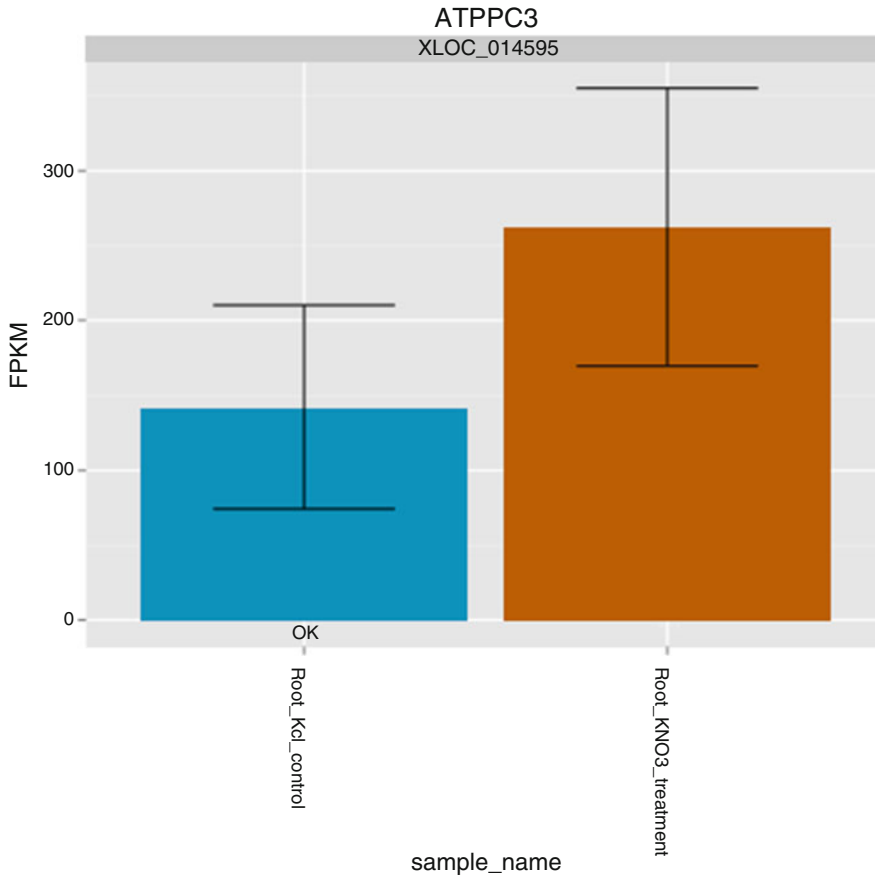


Fig. 5 A bar plot showing the expression of the gene *ATPPC3* under treated (*orange*) and untreated (*blue*) conditions. The average expression levels almost correspond to the pattern indicated in the paper, but with different error bars

seq reads, it is very important to note that certain design aspects of the experiment must be considered in order to eliminate problems in mapping reads and estimating transcript abundance. One of these aspects is to have at least three biological replicates for each condition in order to account for biological variability. Another aspect would be a preference for paired end sequencing instead of single end sequencing, in spite of the higher costs, as TopHat works best with 75 bp reads or longer (up to 1024 bp) from single or paired end fragments. Such conditions considerably improve the alignment and splice junction discovery performance for TopHat, as well as the abundance estimates for Cufflinks.

3. To verify if all the tools were installed correctly, users can visit the following pages and refer to their instructions in the “Testing the Installation” section:

<http://cufflinks.cbcb.umd.edu/tutorial.html>

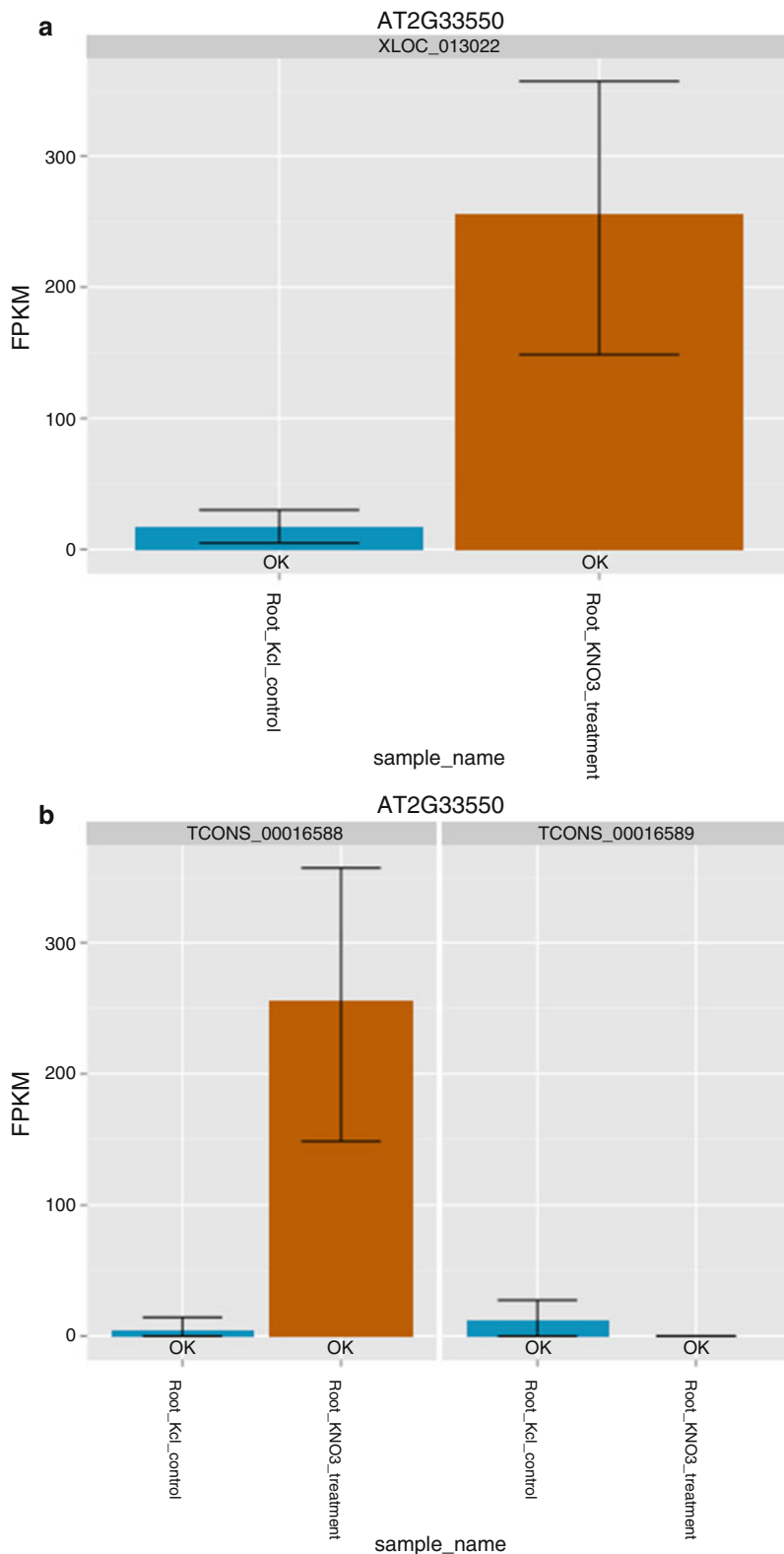


Fig. 6 (a) A bar plot showing the expression of the gene *AT2G33550* under both treated (*orange*) and untreated (*blue*) conditions. **(b)** Bar plots showing the expression levels of the isoforms of the gene *AT2G33550* under both treated (*orange*) and untreated (*blue*) conditions. Note how the isoform FPKM values simply add up to give the total gene expression values for **(a)**

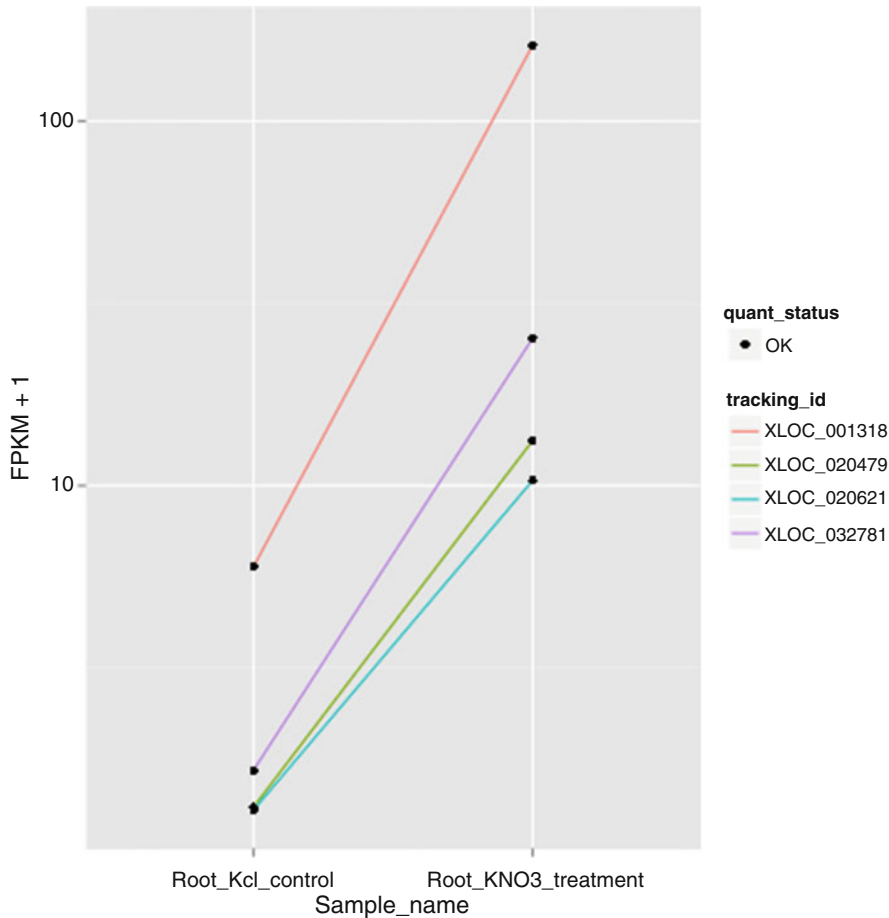


Fig. 7 An illustration output indicating genes that have similar expression patterns to the *BT1* gene

<http://tophat.cbcb.umd.edu/tutorial.shtml>

4. It is very important to note that the CummeRbund package depends on a number of other packages like ggplot2 and RSQLite. A complete list of these packages can be found in the cummeRbund website in the Requirements section (http://compbio.mit.edu/cummeRbund/manual_2_0.html). Most of these packages will automatically be installed as dependencies when the biocLite ('cummeRbund') command is passed. Sometimes, it is advisable to type the command `biocLite()` before installing the cummeRbund package. To get the best results, install the latest version of R and all the dependent packages.
5. The general command structure for running TopHat with single reads is:


```
$ tophat [options] <genome_index_base> <Single_reads1.fq, Single_reads2.fq...>
```

In the example provided, ‘genome’ is the base name of the genome index. The `-i` and `-I` options are for specifying the minimum and maximum intron length respectively (in base pairs). The values of the intron lengths are adjusted as 20 bp and 5000 bp respectively for *Arabidopsis thaliana* [38, 39] instead of the default values which are for mammals. The `-G` option specifies the annotation file as a guide for read mapping. The `-p` option specifies the number of threads to be used for running. We have used single reads instead of paired-end reads. For paired end reads, each end of a pair must be separately included and added pairwise serially as shown below:

```
$ tophat [options] <genome_index_base> <[PEreads1_1.fq,
PEreads2_1.fq..]> <[PEreads1_2.fq, PEreads2_2.fq...]>
```

TopHat version 2.0.10 allows mixing the FASTQ and FASTA file formats. It also allows the input of both single and paired end reads together in one command. The general command is:

```
$ tophat [options]* <genome_index_base> <[PEreads1_1.fq,
PEreads2_1.fq., SEreads.fq]> <[PEreads1_2.fq,
PEreads2_2.fq..]>
```

Or

```
$ tophat [options]* <genome_index_base> <[PEreads1_1.fq,
PEreads2_1.fq..]> <[PEreads1_2.fq, PEreads2_2.fq..,
SEreads.fq]>
```

However, it is not recommended to mix different file types or different types of reads in a single run. In order to run it like this, TopHat has a special protocol advice given in the TopHat manual.

Removing the `-G` option, allows users to quantify the samples without any reference annotation.

In order to quantify the reference annotation only, users need to add the `--no-novel-juncs` option to turn off novel transcript discovery.

For colorspace reads, one has to add the `--bowtie1` option as bowtie2 does not support colorspace. (TopHat assumes that the reads are from an Illumina or SOLiD sequencing machine.)

It is also important to note that TopHat has parameters to specify if the library type is strand specific or not. The default setting is for non-strand specific reads, but for strand specific reads the `--library-type` parameter has to be set as ‘fr-firststrand’ or ‘fr-secondstrand’ according to the protocol followed.

The more advanced options include detection of fusion transcripts using the `--fusion-search` option, and also for using annotation data supplied from another source.

For a comprehensive listing and explanation of all the options available for TopHat, please refer to the TopHat manual using the `tophat --help` command or the manual section of the TopHat website (<http://tophat.cbcb.umd.edu/manual.shtml>).

6. The general command line structure for running Cufflinks is:
`$ cufflinks [options] <accepted_hits.sam/bam>`

Here the `-o` option specifies the output directory for the written files. The TopHat run produces a file called “accepted_hits.bam”. A BAM file is a compressed binary version of SAM. The latter is a read alignment format that can be used very flexibly in many ways. The SAM tools are installed for access and manipulation of the BAM files. Cufflinks uses this output file to assemble the transcripts and isoforms. Like TopHat, Cufflinks has many options which are out of the scope of this example but can be accessed and viewed in the Cufflinks manual (<http://cufflinks.cbcb.umd.edu/manual.html#cufflinks>).

7. The output folder of each run of Cufflinks has a “transcripts.gtf” file, containing information about the assembled transcripts in a tabulated format. The names of each column, their meanings and examples are given in http://cufflinks.cbcb.umd.edu/manual.html#cufflinks_output. These “transcript.gtf” files of each sample and condition are pooled in together in the `assembled_tc.txt` file and Cuffmerge uses this file to search out each transcript file and merge it with the others.

Cuffmerge also performs the functions of Cuffcompare when it runs—Cuffcompare simply compares the transcripts that have been assembled through Cufflinks to a reference annotation file (`.gtf`). Cuffmerge also filters out the artifactual transfrags.

The general command structure for Cuffmerge is:

```
$ cuffmerge [options] <assembly_GTF_list.txt>
```

The `-g` option provides Cuffmerge with a reference annotation file and the `-s` option provides it with a genome sequence file in FASTA format. The provision of the reference annotation file leads to merging of the transcripts and isoforms from the sample with known isoforms and transcripts in the reference, thereby providing a better quality of assembly.

One can run Cuffcompare before running Cuffmerge. For more details on the options provided by Cuffcompare, please refer to <http://cufflinks.cbcb.umd.edu/manual.html#cuffcompare>.

8. Cufflinks estimates transcript abundances from read counts using the FPKM method. During library construction, a specific narrow size range for reads is selected. This means that a transcript twice as long as another will have twice as many reads when their abundances are almost identical. In order to

avoid any kind of misinterpretation with regard to abundance of each from such data, a normalization is performed on the read count of the fragments by dividing it by the total length of the transcript. In another scenario, it is possible that different experiment runs will produce different amounts of reads. So the same transcript will have different abundances in each run. To avoid such kinds of misinterpretations, another normalization step is involved in which the normalized fragment counts are further divided by the unit total fragment hits. Thus, FPKM (or RPKM), short for **f**ragments (or **r**eads) **p**er **k**ilobase of transcript per **m**illion mapped fragments is used which accounts for both the normalization steps. This also makes calculation of gene expression levels easier: Cufflinks calculates abundance estimates for each isoform. So for gene expression determination, one simply needs to add up the FPKM values of each isoform, as FPKM is directly proportional to abundance. In fact, this is also how abundances of transcripts sharing a particular TSS or promoter can be calculated.

9. Cuffdiff calculates differential expression of genes and transcripts across various conditions or time points.

The general command structure for Cuffdiff is:

```
$ cuffdiff [options]<transcripts.gtf><comma_separated_list_of_all_the_accepted_hits.bam_files_in_the_tophat_output_of_each_replicate_for_condition_1><comma_separated_list_of_all_the_accepted_hits.bam_files_in_the_tophat_output_of_each_replicate_for_condition_2>...<..condition N>
```

In this example, the `-o` option specifies the output directory where the Cuffdiff analysis output is to be stored (*diff_result*). The `-p` option specifies the number of threads to be used and the `-b` option specifies that Cuffdiff should use the `genome.fa` file to make corrections for biases and improve abundance estimate accuracies. The `-u` option is an instruction to accurately weight the reads that map to multiple locations in the genome. The `-L` options specifies the labels that Cuffdiff has to assign to each sample condition in the arguments. Cuffdiff requires a GTF file, produced by Cuffcompare or Cuffmerge (or any other source where the chromosome names must match those in the SAM/BAM alignment files provided) as an argument. The second argument is the alignment files for each sample. Replicates of each sample should be included in the sample argument as comma separated values.

Cuffdiff output contains a set of files containing various information about the differential splicing and expression events across the samples.

For more advanced options and details of Cuffdiff output files, please refer to the Cuffdiff manual in the Cufflinks website (<http://cufflinks.cbcb.umd.edu/manual.html#cuffdiff>).

10. You are likely to get an error in the creation of the ‘cuffdata’ object. In such case it is better that the hashes be removed from the merged.gtf file produced by Cuffmerge before running Cuffdiff on it.
11. The plots usually are all recorded in a file called Rplots.pdf in the working directory. Often this is not convenient for users. In which case it is better for users to follow the given set of commands:

```
>pdf(file = 'file_name.')
>command_for_creating_visualization
>dev.off()
```

For example, if you were to save a density plot as a pdf with a particular file name, you would type the following commands:

```
>pdf(file = 'scatterplot.pdf')
>csScatter(genes(cuffdata), 'C1', 'C2')
>dev.off()
```

Alternatively, you can save the images in another format like tiff in the following way:

```
>tiff(file = 'file_name.tiff')
>command_for_plot
>dev.off()
```

References

1. Ozsolak F, Milos PM (2011) RNA sequencing: advances, challenges and opportunities. *Nat Rev Genet* 12(2):87–98. doi:[10.1038/nrg2934](https://doi.org/10.1038/nrg2934)
2. Metzker ML (2010) Sequencing technologies - the next generation. *Nat Rev Genet* 11(1):31–46. doi:[10.1038/nrg2626](https://doi.org/10.1038/nrg2626)
3. Mardis ER (2008) The impact of next-generation sequencing technology on genetics. *Trends Genet* 24(3):133–141. doi:[10.1016/j.tig.2007.12.007](https://doi.org/10.1016/j.tig.2007.12.007)
4. Marioni JC, Mason CE, Mane SM, Stephens M, Gilad Y (2008) RNA-seq: an assessment of technical reproducibility and comparison with gene expression arrays. *Genome Res* 18(9):1509–1517. doi:[10.1101/gr.079558.108](https://doi.org/10.1101/gr.079558.108)
5. Ozsolak F, Platt AR, Jones DR, Reifengerber JG, Sass LE, McInerney P, Thompson JF, Bowers J, Jarosz M, Milos PM (2009) Direct RNA sequencing. *Nature* 461(7265):814–818. doi:[10.1038/nature08390](https://doi.org/10.1038/nature08390)
6. Roy SW, Irimia M (2008) When good transcripts go bad: artifactual RT-PCR ‘splicing’ and genome analysis. *BioEssays* 30(6):601–605. doi:[10.1002/bies.20749](https://doi.org/10.1002/bies.20749)
7. Garber M, Grabherr MG, Guttman M, Trapnell C (2011) Computational methods for transcriptome annotation and quantification using RNA-seq. *Nat Methods* 8(6):469–477. doi:[10.1038/nmeth.1613](https://doi.org/10.1038/nmeth.1613)
8. Trapnell C, Williams BA, Pertea G, Mortazavi A, Kwan G, van Baren MJ, Salzberg SL, Wold BJ, Pachter L (2010) Transcript assembly and

- quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat Biotechnol* 28(5):511–515. doi:[10.1038/nbt.1621](https://doi.org/10.1038/nbt.1621)
9. Wang Z, Gerstein M, Snyder M (2009) RNA-Seq: a revolutionary tool for transcriptomics. *Nat Rev Genet* 10(1):57–63. doi:[10.1038/nrg2484](https://doi.org/10.1038/nrg2484)
 10. Schena M, Shalon D, Davis RW, Brown PO (1995) Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science* 270(5235):467–470
 11. Schena M, Shalon D, Heller R, Chai A, Brown PO, Davis RW (1996) Parallel human genome analysis: microarray-based expression monitoring of 1000 genes. *Proc Natl Acad Sci U S A* 93(20):10614–10619
 12. Shiraki T, Kondo S, Katayama S, Waki K, Kasukawa T, Kawaji H, Kodzius R, Watahiki A, Nakamura M, Arakawa T, Fukuda S, Sasaki D, Podhajska A, Harbers M, Kawai J, Carninci P, Hayashizaki Y (2003) Cap analysis gene expression for high-throughput analysis of transcriptional starting point and identification of promoter usage. *Proc Natl Acad Sci U S A* 100(26):15776–15781. doi:[10.1073/pnas.2136655100](https://doi.org/10.1073/pnas.2136655100)
 13. Kodzius R, Kojima M, Nishiyori H, Nakamura M, Fukuda S, Tagami M, Sasaki D, Imamura K, Kai C, Harbers M, Hayashizaki Y, Carninci P (2006) CAGE: cap analysis of gene expression. *Nat Methods* 3(3):211–222. doi:[10.1038/nmeth0306-211](https://doi.org/10.1038/nmeth0306-211)
 14. Velculescu VE, Zhang L, Vogelstein B, Kinzler KW (1995) Serial analysis of gene expression. *Science* 270(5235):484–487
 15. Reinartz J, Bruyins E, Lin JZ, Burcham T, Brenner S, Bowen B, Kramer M, Woychik R (2002) Massively parallel signature sequencing (MPSS) as a tool for in-depth quantitative gene expression profiling in all organisms. *Brief Funct Genomic Proteomic* 1(1):95–104
 16. Brenner S, Johnson M, Bridgham J, Golda G, Lloyd DH, Johnson D, Luo S, McCurdy S, Foy M, Ewan M, Roth R, George D, Eletr S, Albrecht G, Vermaas E, Williams SR, Moon K, Burcham T, Pallas M, DuBridg RB, Kirchner J, Fearon K, Mao J, Corcoran K (2000) Gene expression analysis by massively parallel signature sequencing (MPSS) on microbead arrays. *Nat Biotechnol* 18(6):630–634. doi:[10.1038/76469](https://doi.org/10.1038/76469)
 17. Mortazavi A, Williams BA, McCue K, Schaeffer L, Wold B (2008) Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nat Methods* 5(7):621–628. doi:[10.1038/nmeth.1226](https://doi.org/10.1038/nmeth.1226)
 18. Feldmeyer B, Wheat CW, Krezdorn N, Rotter B, Pfenninger M (2011) Short read Illumina data for the de novo assembly of a non-model snail species transcriptome (*Radix balthica*, Basommatophora, Pulmonata), and a comparison of assembler performance. *BMC Genomics* 12:317. doi:[10.1186/1471-2164-12-317](https://doi.org/10.1186/1471-2164-12-317)
 19. Langmead B, Trapnell C, Pop M, Salzberg SL (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol* 10(3):R25. doi:[10.1186/gb-2009-10-3-r25](https://doi.org/10.1186/gb-2009-10-3-r25)
 20. Li H, Durbin R (2010) Fast and accurate long-read alignment with Burrows-Wheeler transform. *Bioinformatics* 26(5):589–595. doi:[10.1093/bioinformatics/btp698](https://doi.org/10.1093/bioinformatics/btp698)
 21. Simpson JT, Durbin R (2010) Efficient construction of an assembly string graph using the FM-index. *Bioinformatics* 26(12):i367–i373. doi:[10.1093/bioinformatics/btq217](https://doi.org/10.1093/bioinformatics/btq217)
 22. Trapnell C, Pachter L, Salzberg SL (2009) TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics* 25(9):1105–1111. doi:[10.1093/bioinformatics/btp120](https://doi.org/10.1093/bioinformatics/btp120)
 23. Trapnell C, Roberts A, Goff L, Pertea G, Kim D, Kelley DR, Pimentel H, Salzberg SL, Rinn JL, Pachter L (2012) Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. *Nat Protoc* 7(3):562–578. doi:[10.1038/nprot.2012.016](https://doi.org/10.1038/nprot.2012.016)
 24. Goff L, Trapnell C, Kelley D, Guide PRSCU, bioeViews Clustering D, DataRepresentation D, GeneExpression I, MultipleComparison Q, RNASeq R, since BioC IB (2012) Analysis, exploration, manipulation, and visualization of Cufflinks high-throughput sequencing data. R package version 2 (1)
 25. Goecks J, Nekrutenko A, Taylor J, Galaxy T (2010) Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biol* 11(8):R86. doi:[10.1186/gb-2010-11-8-r86](https://doi.org/10.1186/gb-2010-11-8-r86)
 26. Li R, Yu C, Li Y, Lam TW, Yiu SM, Kristiansen K, Wang J (2009) SOAP2: an improved ultrafast tool for short read alignment. *Bioinformatics* 25(15):1966–1967. doi:[10.1093/bioinformatics/btp336](https://doi.org/10.1093/bioinformatics/btp336)
 27. Lunter G, Goodson M (2011) Stampy: a statistical algorithm for sensitive and fast mapping of Illumina sequence reads. *Genome Res* 21(6):936–939. doi:[10.1101/gr.111120.110](https://doi.org/10.1101/gr.111120.110)
 28. Wang K, Singh D, Zeng Z, Coleman SJ, Huang Y, Savich GL, He X, Mieczkowski P, Grimm SA, Perou CM, MacLeod JN, Chiang DY, Prins JF, Liu J (2010) MapSplice: accurate

- mapping of RNA-seq reads for splice junction discovery. *Nucleic Acids Res* 38(18), e178. doi:[10.1093/nar/gkq622](https://doi.org/10.1093/nar/gkq622)
29. Wu TD, Nacu S (2010) Fast and SNP-tolerant detection of complex variants and splicing in short reads. *Bioinformatics* 26(7):873–881. doi:[10.1093/bioinformatics/btq057](https://doi.org/10.1093/bioinformatics/btq057)
30. Guttman M, Garber M, Levin JZ, Donaghey J, Robinson J, Adiconis X, Fan L, Koziol MJ, Gnirke A, Nusbaum C, Rinn JL, Lander ES, Regev A (2010) Ab initio reconstruction of cell type-specific transcriptomes in mouse reveals the conserved multi-exonic structure of lincRNAs. *Nat Biotechnol* 28(5):503–510. doi:[10.1038/nbt.1633](https://doi.org/10.1038/nbt.1633)
31. Zerbino DR, Birney E (2008) Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome Res* 18(5):821–829. doi:[10.1101/gr.074492.107](https://doi.org/10.1101/gr.074492.107)
32. Lee S, Seo CH, Lim B, Yang JO, Oh J, Kim M, Lee S, Lee B, Kang C, Lee S (2011) Accurate quantification of transcriptome from RNA-Seq data by effective length normalization. *Nucleic Acids Res* 39(2):e9. doi:[10.1093/nar/gkq1015](https://doi.org/10.1093/nar/gkq1015)
33. Wang L, Feng Z, Wang X, Wang X, Zhang X (2010) DEGseq: an R package for identifying differentially expressed genes from RNA-seq data. *Bioinformatics* 26(1):136–138. doi:[10.1093/bioinformatics/btp612](https://doi.org/10.1093/bioinformatics/btp612)
34. Anders S, Huber W (2010) Differential expression analysis for sequence count data. *Genome Biol* 11(10):R106. doi:[10.1186/gb-2010-11-10-r106](https://doi.org/10.1186/gb-2010-11-10-r106)
35. Twine NA, Janitz K, Wilkins MR, Janitz M (2011) Whole transcriptome sequencing reveals gene expression and splicing differences in brain regions affected by Alzheimer's disease. *PLoS One* 6(1), e16266
36. Vidal EA, Moyano TC, Krouk G, Katari MS, Tanurdzic M, McCombie WR, Coruzzi GM, Gutierrez RA (2013) Integrated RNA-seq and sRNA-seq analysis identifies novel nitrate-responsive genes in *Arabidopsis thaliana* roots. *BMC Genomics* 14:701. doi:[10.1186/1471-2164-14-701](https://doi.org/10.1186/1471-2164-14-701)
37. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R, Genome Project Data Processing S (2009) The Sequence Alignment/Map format and SAMtools. *Bioinformatics* 25(16):2078–2079. doi:[10.1093/bioinformatics/btp352](https://doi.org/10.1093/bioinformatics/btp352)
38. Alexandrov NN, Troukhan ME, Brover VV, Tatarinova T, Flavell RB, Feldmann KA (2006) Features of *Arabidopsis* genes and genome discovered using full-length cDNAs. *Plant Mol Biol* 60(1):69–85. doi:[10.1007/s11103-005-2564-9](https://doi.org/10.1007/s11103-005-2564-9)
39. Filichkin SA, Priest HD, Givan SA, Shen R, Bryant DW, Fox SE, Wong WK, Mockler TC (2010) Genome-wide mapping of alternative splicing in *Arabidopsis thaliana*. *Genome Res* 20(1):45–58. doi:[10.1101/gr.093302.109](https://doi.org/10.1101/gr.093302.109)

INDEX

A

- ACE file format.....254, 255
- AFG file format254
- Algorithmics on words294
- Amino acid sequence..... 29, 35, 52, 67, 154, 176, 195
- Annotation tracks.....207, 264–265
- Assemblies..... 6, 90, 107, 136, 143, 168, 189, 195, 254, 257, 276
- Assessing reference sequence contiguity.....258–261

B

- BAM file format 118, 125, 158, 160, 161, 254–256, 258, 261, 262, 264–268, 271, 276, 325, 357, 358
- Barley genome..... 119, 143, 144, 167, 171, 174, 176, 234, 237
- Basic Local Alignment Search Tool (BLAST).....5, 8, 9, 11–15, 18, 19, 21, 26, 28, 30, 32–34, 52, 67, 136, 147, 169, 175, 178, 181, 182, 184, 193–196, 204, 205, 208, 209, 211, 215, 224–228, 300–305, 307, 310, 312, 313, 322, 324
- BED file format..... 118, 158, 254
- Biocuration..... 35, 77–78
- Bioinformatics 23, 99, 106, 160, 265, 271, 275, 296, 319
- Biological pathways55
- BioMart.....117, 121, 128, 130–132
- BLAST like alignment tool (BLAT)..... 122, 302, 303
- Bowtie302, 303, 340, 341, 343, 344
- Burrows-Wheeler Transform (BWT) 242, 243, 275, 319, 320, 322

C

- Cereals..... 119, 120, 171, 176
- Comparative genomics89, 91, 116, 122–123, 138, 141, 154, 218, 307
- Comparative map 47, 209
- Comparative pathways analysis141–161
- Copy number variant (CNV)250
- Crops..... 115, 116, 119, 120, 141, 166, 176, 203, 204, 215, 234, 270
- CrowsNest synteny browser.....176
- Cuffcompare..... 342, 357, 358
- Cuffdiff.....342, 347, 348, 358, 359

- Cufflinks..... 181, 340–342, 345, 346, 352, 357, 359
- Cuffmerge 342, 346, 357–359
- CummeRbund.....340, 342, 345, 347, 355

D

- Databases..... 2–7, 11–13, 18, 31, 47, 57, 60, 75, 90, 106, 107, 116–117, 121, 128, 135, 146, 171, 197, 199–200, 206, 295
- Differential gene expression342, 343, 347–350, 358
- DNA sequence 7, 138, 158, 178, 187, 190, 194, 230, 233, 241, 242, 294

E

- Entrez.....2, 7–8, 10–12, 14–20

F

- FASTA sequence format 6, 9, 12–14, 19–21, 42, 67, 81, 127, 131, 135, 138, 155, 178, 192, 195, 208, 234, 237, 239, 240, 244, 254, 255, 286, 325, 344, 356, 357
- Fragments per kilobase of exon per million fragments mapped (FPKM) 347, 352, 354, 357
- Functional genomics..... 75, 116, 141, 340

G

- Gapped alignment250
- Gene homology.....7, 146, 178, 234, 237
- Genetics..... 47, 94, 106, 129, 146, 157, 188–191, 194, 295
 - map..... 187, 190, 193, 196, 198, 205, 209–211, 278, 280–282, 285
 - variation.....157
- Genome6, 11, 47, 55, 64, 71–73, 89, 106, 107, 111, 115–118, 122–127, 130, 131, 135, 142–145, 147, 148, 150, 154, 160, 168, 172–174, 176, 178–179, 187, 189, 190, 192, 195–196, 199, 206, 217, 229, 271, 295–313, 343
 - annotation.....47, 64
 - browser 117, 123–126, 136–138, 142, 147, 148, 150–154, 158, 160, 161, 187, 189, 190, 192, 195–196, 199, 206
 - indexing.....243, 356
 - viewer 278, 279

GenomeZipper.....174–176
 Genomic repeats.....295, 319
 Genomics47, 48, 57, 94, 106, 122–123,
 146, 158, 188–190, 225, 313–325, 328
 Genotyping285
 Genotyping by sequencing.....285
 GFF3 genome feature format.....127, 131, 254, 257,
 264, 265, 267
 Gramene database73, 76, 154

H

Hash table243

I

Illumina.....166, 170, 204, 208, 233, 234, 244,
 258, 259, 264, 270–272, 286, 294, 316, 343, 356
 Indexing314–316, 319–322
 Integrated database.....24, 56, 184, 205, 228
 Intron-exon structure.....264–265

K

KEGG mapper.....67, 68
 KEGG pathway map.....56, 57, 61–63, 67, 68
 KnowledgeBase23–53, 204

M

Maize.....47, 94, 106, 119, 176,
 188–191, 194, 198
 MaizeCyc72, 74, 79, 146, 191, 197
 Manual annotation.....23–53
 MapMan71, 73–77, 80–82, 84, 85, 198
 Mapping.....249, 264, 265, 276
 Match mode.....245
 MegaBLAST12, 208, 234, 240
 Metabolic pathways.....191, 197
 Metabolism58, 61
 Misassemblies.....262
 Mismatches243–246, 248–251, 299,
 321, 330–332
 Model organism database.....94, 237
 Molecular interactions.....56, 59, 71, 146
 Mutation5, 18, 58, 187, 243, 246

N

National Center for Biotechnology Information
 (NCBI).....2–9, 11, 12, 14–21, 33, 36,
 90, 146, 169, 187, 190, 195, 237, 240, 261, 301, 303,
 343, 352
 Next-generation sequencing (NGS).....4, 14, 166,
 167, 170, 172, 173, 181, 190, 203, 241–244, 253–273,
 275, 278, 294, 299, 303, 304, 321, 324, 325, 339, 340
 Nucleotides.....2, 7, 10, 15, 237, 259, 285

O

Ontology24, 35, 48, 90–93, 95–103, 105,
 106, 109–111, 118, 121, 128, 130–131, 137, 150, 182,
 191, 218, 226
 Orthologous sequences.....158, 167, 276

P

Paired end reads.....174, 243, 246, 250, 261, 270,
 277, 279, 342, 356
 Pattern matching.....296, 304, 314, 325, 327–332
 Phenomics.....89
 Phenotype122, 192
 Phylogenetics.....4, 5, 74, 128, 129, 150,
 154–156, 172, 173, 303
 Phytochemical compounds.....61
 Plant anatomy91, 102
 Plant development.....93, 102, 109
 Plant gene expression150
 Plant genome.....177
 Plant genome annotation109
 Plant metabolic pathway56, 74, 150
 Plant metabolism.....69
 Plant ontology (PO).....89–110, 118, 128, 137, 191
 Plant pathway databases71–85, 142
 Plant pathways62–64, 74, 75, 77, 85
 Plant regulatory pathway.....145, 150
 PlantsDB.....165, 171, 176–178, 181
 Promoter discovery.....233
 Protein database7

R

Reactome.....71, 73, 145, 147, 149, 160
 Read data.....259
 Read mapping249, 273
 Reference genomes143
 Regulatory pathways.....71, 145, 150
 Repeats121, 171, 295, 298, 301–303,
 306–308, 310, 312–325, 327, 332
 Ribonucleic acid sequencing (RNA-seq).....171, 172,
 174, 181, 190, 197, 327, 329, 340–343, 345, 352–353
 RiceCyc.....72, 74, 79, 146

S

SAM file format.....246, 254, 255, 258, 265, 271,
 272, 276, 281, 344–345, 357, 358
 Second generation sequencing (SGS).....203, 233, 234
 Seed length.....248–249
 Semantic web99
 Sequence analysis26
 Short Oligonucleotide Analysis Package
 (SOAP).....19, 243, 244, 246, 247, 249, 276
 Short read alignment.....242

Signaling pathways71
 Single nucleotide polymorphism (SNP) 7, 122, 146,
 152, 157, 158, 187–190, 204, 207, 211, 212, 241, 243,
 249–251, 258, 259, 266, 267, 270, 271, 276–278, 280,
 282, 285, 286, 288–291, 324, 325
 calling249, 250, 276, 277, 288, 291
 discovery271
 prediction241
 String data structures296
 Structural variation118, 125, 141, 146, 154,
 190, 241, 307, 339, 341
 Swiss-Prot24–26, 31, 34–36, 45,
 49, 53, 211
 Synteny215

T

TAGdb205, 208, 209, 234–240
 TopHat290, 340–346, 352, 355–357
 Transcriptome assembly 122, 341, 342
 Transcriptomics 67, 340
 Transport pathways145
 Transposable elements294

Transposons303, 310
 TrEMBL 24, 26, 34, 35, 45, 47, 49, 53
Triticum aestivum 4, 120, 144, 166, 203, 237,
 239, 240, 294

U

Ungapped alignment 243, 244, 251
 UniProt23–26, 28–34, 36, 42, 48,
 51–53, 57, 128, 204

V

Variants 118, 121, 130, 136, 146,
 257–259, 267, 271, 281
 VCF file format 118, 131, 138, 158, 254,
 271, 277, 278, 288
 Visualization265

W

Wheat genome 121, 165–174, 203–212
 WikiPathways74, 77–79, 82–83, 85
 Word index 315, 316, 332