Chapter 3
Information Retrieval in Life Sciences: A Programmatic Survey

Matthias Lange, Ron Henkel, Wolfgang Müller, Dagmar Waltemath, and Stephan Weise

Abstract Biomedical databases are a major resource of knowledge for research in the life sciences. The biomedical knowledge is stored in a network of thousands of databases, repositories and ontologies. These data repositories differ substantially in granularity of data, storage formats, database systems, supported data models and interfaces. In order to make full use of available data resources, the high number of heterogeneous query methods and frontends requires high bioinformatic skills. Consequently, the manual inspection of database entries and citations is a time-consuming task for which methods from computer science should be applied.

Concepts and algorithms from information retrieval (IR) play a central role in facing those challenges. While originally developed to manage and query less structured data, information retrieval techniques become increasingly important for the integration of life science data repositories and associated information. This chapter provides an overview of IR concepts and their current applications in life sciences. Enriched by a high number of selected references to pursuing literature, the following sections will successively build a practical guide for biologists and bioinformaticians.

M. Lange (✉) • S. Weise
Leibniz Institute of Plant Genetics and Crop Plant Research, Bioinformatics and Information Technology, OT Gatersleben, Corrensstraße 3, 06466 Stadt Seeland, Germany
e-mail: lange@ipk-gatersleben.de; weise@ipk-gatersleben.de

R. Henkel • D. Waltemath
Department of Systems Biology and Bioinformatics, University of Rostock, Ulmenstraße 69, 18057 Rostock, Germany
e-mail: ron.henkel@uni-rostock.de; dagmar.waltemath@uni-rostock.de

W. Müller
HITS gGmbH, Schloss-Wolfsbrunnenweg 35, 69118 Heidelberg, Germany
e-mail: wolfgang.mueller@h-its.org

M. Chen and R. Hofestädt (eds.), *Approaches in Integrative Bioinformatics: Towards the Virtual Cell*, DOI 10.1007/978-3-642-41281-3_3,

Keywords Information retrieval • Data management • Search engines • Relevance ranking • Recommender systems • Semantic data networks • Data integration

3.1 Motivation: Information Systems in Life Sciences

The progress in molecular biology, ranging from experimental data acquisition on individual genes and proteins, over postgenomic technologies, such as RNA-seq, phenotyping, proteomics, systems biology and integrative bioinformatics aims to capture the big picture of entire biological systems [55]. As a consequence of this revolution, the amount of data in the life sciences has exploded. The wave of new technologies, for example, in genomics, is enabling data to be generated at unprecedented scales [85]. As of February 2013, NCBI GenBank provides access to 162,886,727 sequences, and PubMed comprises over 22 million citations for biomedical literature from MEDLINE, life science journals and online books. The number of public available databases passed recently the high water mark of 1,512 [32]. This data deluge must now be harnessed and exploited.

Another aspect is the continuous developments in information procurement, preparation and processing as shown in Fig. 3.1. Over the past years, information processing techniques evolved from library research and individual data archives to web-based systems using intercontinental high-speed network links for an ad hoc data exchange, cloud computing and distributed databases. This continuous and

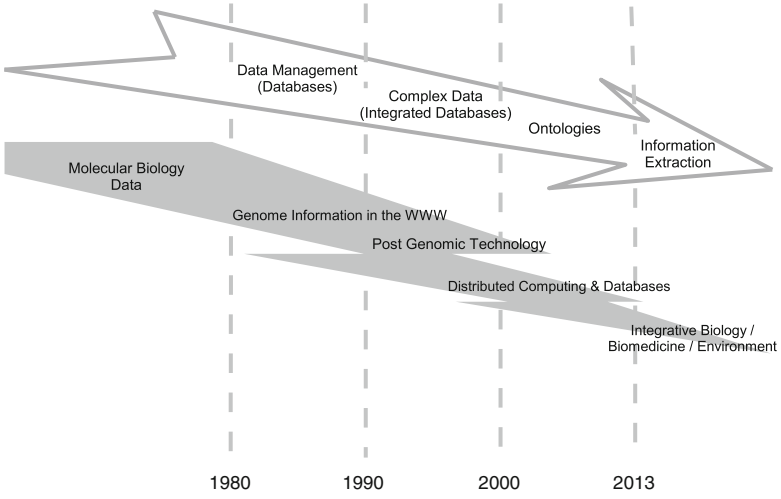

Fig. 3.1 The development of information processing in life sciences adapted from [101] (Reprinted by permission from Macmillan Publishers Ltd, copyright 2002) – Classic database management systems and the domain-specific modelling of project databases are replaced by integrative technologies, i.e. data warehouses, data networks and information retrieval

ongoing shift is attended by the use of *database management systems (DBMS)* which are applied to the management of increasingly complex data structures and voluminous content [98]. The key concepts in bioinformatics with regard to data handling are a consistent classification and unambiguous definition of the modelled biological objects in the databases, the raising use of ontologies, connected with methods of knowledge processing, information extraction and data mining [82, 97].

The consequences of this development are new requirements for information retrieval methods. Typically, life scientists and bioinformaticians formulate their queries rather vaguely. This does not necessarily happen due to inexperience or ignorance but because their search is often explorative with no clear idea of the expected answer. Vague queries though pose a problem on current databases and information systems as these queries cannot be semantically interpreted, without comprehensive semantic document tagging or the use of controlled vocabulary. Much more specific problems such as data distribution and isolation, structural heterogeneity, less metadata, interfaces query languages and deep (invisible) web are further examples of the underlying challenges.

In this context, *information retrieval (IR)* is getting increased importance as technology to face heterogeneities in data representation, storage and organisation towards an efficient information access. The methods for representation and organisation of information items should be designed in accordance to provide users an easy access to the information of their interest [8]. The first step towards this formulated aims is a raising need to find, extract, merge and synthesise information from multiple, disparate sources [56]. In particular, the convergence of biology, computer science and information technology will accelerate this multidisciplinary endeavour. The basic needs for IR are summarised in [58]:

1. On-demand access and retrieval of the most up-to-date biological data and the ability to perform complex queries across multiple heterogeneous databases to find the most relevant information
2. Access to the best-of-breed analytical tools and algorithms for extraction of useful information from the massive volume and diversity of biological data
3. A robust information integration infrastructure that connects various computational steps involving database queries, computational algorithms and application software

Information retrieval in life science databases exhibits some fundamental differences from the way people search in the web or in a general-purpose digital library. First of all, links play a central role for data integration. Not only a single article to a specific entity is of relevance, but all linked articles may be relevant. However, articles just mentioning the entity of relevance may be irrelevant. Second, life science databases are organised in a domain-centric manner, usually concentrating around specific entity types (e.g. metabolomics). It is easy to extract all domain information related to one entity. In contrast, it is very difficult to collect comprehensive, cross-domain information on an entity if the knowledge is spread across entities of different domains, e.g. genome structure-focused databases

versus metabolite or pathway-centric ones. A similar picture of heterogeneity can be observed in data access and querying. Methods spread among Boolean queries; predefined queries in web information systems, also known as canned queries; semantic web; keyword-based retrieval in text documents; relevance ranking; and recommender systems are commonly used in life science dry labs.

In this chapter, we will subsequently introduce relevant concepts for information retrieval in the life sciences. It is organised as follows: The Sect. 3.2 provides an overview of basic concepts for data storage, metadata formats and query interfaces, as well as data integration. The Sect. 3.3 then introduces the theoretical foundations, the core concepts of information retrieval and the specific implementation in life sciences. Here, the focus is on characteristics of information retrieval in the life sciences, exploratory information retrieval, recommender systems, human–computer interfaces and semantic aspects with an emphasis on model databases and data networks. The life science search engine LAILAPS is presented as example for an exploratory IR system. The last section contains a comprehensive summary of this chapter.

3.2 Information Systems and Databases

In general, the term *information system (IS)* describes a somehow connected compound of information [89]. In computer science, an information system aims, manages and provides information to support all necessary processes and workflows, especially in companies. Usually, an information system consists of different applications, which are interacting with a *database management system (DBMS)*. Information systems are a main focus in business information technology.

In computer science, a *database (DB)* is a well-structured and functionally associated set of data [29]. A database is managed by a special software – the so-called database management system (DBMS). Together, DB and DBMS form a *database system (DBS)*. The majority of database systems are using the *relational database model* [18].

In life sciences, the term database is often used as a synonym for the term information system. Since the data volume in life sciences is growing rapidly [82], e.g. due to high-throughput technologies (see also Sect. 3.1), the importance of information systems in this area of research is increasing continuously. Often information systems in life sciences use a data basis that is not organised in database management systems [17], but flat files, markup files, HTML or XML files instead. Moreover, the systems are specific to only one data domain. A third characteristic of information systems in life sciences is that they provide different means of access, e.g. web interfaces, web services or static HTML pages, and provide different data exchange formats. The resulting challenges will be described in the following sections.

3.2.1 Data Domains

A *data domain* comprises all data of a specific area, e.g. the domain of the sequence data or the domain of the phenotypic data. Even though data domains can be analysed separately, a combined analysis of multiple data domains, e.g. genotype–phenotype correlations, provides a much higher chance for success. Subsequently, some examples of data domain are listed. Without the intention of providing a comprehensive classification of life science data domains, this list will give an impression about their wide range and diversity.

- *Sequence data:* In biology, this term refers to sequences of nucleotides (DNA sequence) or sequences of amino acids (amino acid sequence/protein sequence), which are the result of a sequencing. Here, sequencing means the determination of all sub-elements. Several sequencing technologies have been developed. Examples are "classical" techniques, such as Sanger sequencing (chain-termination method) [84], Maxam–Gilbert sequencing [73] or EST-based sequencing [2], and next-generation sequencing (NGS) techniques, such as 454 pyrosequencing [72] or Illumina (Solexa) sequencing [11].
- *Variation and marker data:* In genetics, a marker is a piece of DNA with a known location in the genome, which has different expressions in different organisms. Examples are restriction fragment length polymorphism (RFLP) markers [13] or single nucleotide polymorphism (SNP) markers [103]. Today, large amounts of marker data can be obtained by high-throughput technologies.
- *Expression data:* Gene expression means the transformation of DNA information into structures or functions of cells, e.g. the synthesis of enzymes. Depending on different criteria, such as special tissues or compartments, developmental stages or environmental effects, varying amounts of gene products are produced (expressed). With array technologies [86] or by help of RNA-seq, a multitude of product concentrations can be analysed simultaneously (expression profiling).
- *Metabolic network data:* Metabolic networks (pathways) are sequences of biochemical reactions. They can be different depending on the organism, developmental stages, subcellular loci, etc. Data about these networks is an important basis for the understanding of biological subjects at a systems level [104].
- *Phenotypic data:* The phenotype of an organism comprises all characteristics (traits) which can be observed directly and indirectly. It covers a large variety of traits. Besides traits that are mostly determined genetically (e.g. the hair colour), there are also many traits which depend on environmental effects, such as biotic or abiotic stresses.
- *Passport data:* Not often used in the "classical" bioinformatics, but for the management of *plant genetic resources (PGR)* in the so-called gene banks, passport data is indispensable. Passport data contains information, which is used to uniquely identify genotypes.

- *Literature data:* In science, the structured management of literature references is of high importance. Central databases, such as NCBI PubMed[1] or DBLP,[2] collect millions of references from thousands of journals, proceedings, etc. and provide this data to the scientific community. Such information is often used for text mining approaches.

3.2.2 Data Interfaces and Query Methods

Data is only useful if it can be found on request. Consequently, appropriate *query mechanisms* are a prerequisite to reusing existing knowledge in databases. In this respect, queries should be independent from the physical data format, and it should be possible to extract data by specific criteria or to perform database operations, respectively. For performing database operations, query languages can be used, which are based on a data model. Here, it can be distinguished between *procedural* and *declarative* query languages. The former case can be implemented using sequential programming or nesting of database operators, whereas in the latter case only the structure of the results needs to be defined. In other words, only the "what" will be specified, but not the "how".

Data interfaces are necessary for linking applications and data management. These interfaces can be implemented as so-called *application programming interfaces (APIs)*. Common communication interfaces for linking applications and databases are:

- *(Local) File-based access:* A simple method to access data is the use of files from a local file system. This also includes network file systems, e.g. NFS, and file access via data transfer protocols, e.g. FTP. For the data access, the whole file must be parsed. Since the data format is known, data elements can be extracted and then be transferred into data structures. Several parsers have been implemented and are available via APIs (see Sect. 3.2.3).
- *Remote procedure call (RPC):* Another possibility for accessing data is the call of distant (or remote) methods. These comprise protocols such as REST,[3] SOAP,[4] DCOM [16], .NET or CORBA [93]. These methods provide extended functionality, ranging from simple method calls to distributed object networks, web services or persistence frameworks. An essential feature of these standards is the independence of programming languages.
- *DBMS query APIs:* A combination of data query languages and APIs enables remote data access, similarly to DBMS functionality. The technology behind

[1] http://www.ncbi.nlm.nih.gov/pubmed/

[2] http://dblp.uni-trier.de/

[3] http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

[4] http://www.w3.org/TR/soap/

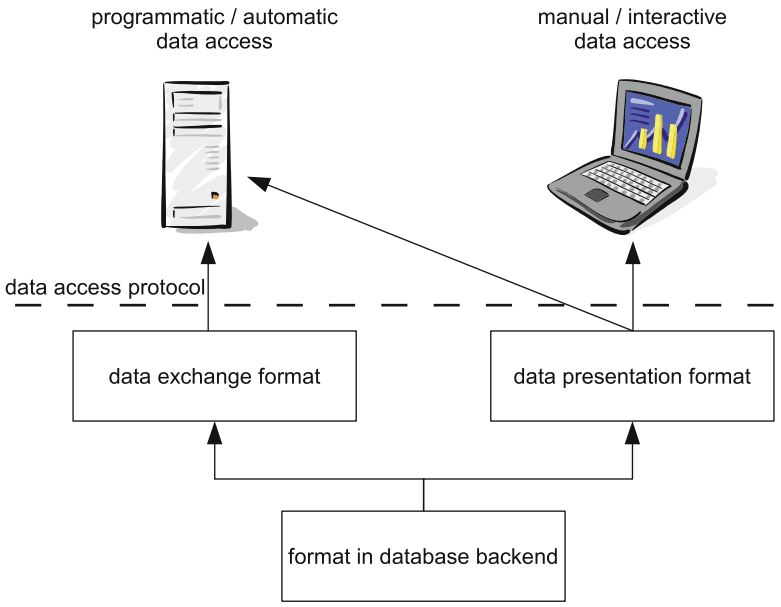

Fig. 3.2 Abstract schema to data storage and format layer in life sciences

either embeds special database access commands into the programming language or integrates the data query language with function calls using APIs. Here already existing programming language-specific APIs and DBMS-specific APIs can be reused. Moreover, DBMS abstracting architectures, such as JDBC [94] or ODBC [33], are available.

3.2.3 Data Formats

A *data format* is a well-defined structure to persistently store data in one or more files. File-based data formats are widely used for the exchange and presentation of data in life sciences [1]. The actual data format is dependent on the storage level and the required access patterns. As shown in Fig. 3.2, it is useful to distinguish different storage layers, which are backend, data exchange and data presentation. The backend layer has a particular emphasis on effective persistence and efficient access structures. In contrast, the data exchange layer is focused on supporting a platform-independent format enriched with structural and semantic metadata. The presentation layer is optimised for an optimal layout and should be flexible to support different HCI technologies and devices.

Whether the *data backend* is a DBMS or it is based on flat file techniques, data independence can be assumed. Thus, data formats used here shall not be dealt with

in detail. For *data presentation*, HTML is widely used as a data format. While the content of HTML pages can also be extracted using parsers, however, HTML only plays a minor role for *data exchange*. This is because HTML is mainly used to present and structure elements and the focus is more on the visual layout of data. This hampers the machine-based processing. A more suitable format for data exchange is the Extensible Markup Language (XML).

In addition, the use of domain-specific, not necessarily formal, defined text *flat files* plays an important role. Popular databases use such formats, e.g. EMBL [52]. Another example is the FASTA format [79] which was originally developed for a bioinformatics tool for sequence comparisons. Today, it is a de facto standard for sequence data exchange. A third example is the so-called two-letter code for databases from the European Bioinformatics Institute (EBI) which uses attribute–value pairs.

In the case of flat files, only an indispensable *format description* enables the development of parsers. Such a description should contain the following elements:

- *Allowed constructs:* All allowed words are specified as combinations of valid characters.
- *Syntax description:* The syntax specifies rules for constructing valid combinations, sequences and structures of the constructs described above.
- *Data schema semantics:* Here, rules for mapping the data format structures into elements and relationships of data schemata are specified.

For molecular biological databases, *formal and informal descriptions* of the format are common practice for both, allowed constructs and syntax description. In contrast, data schema semantics are only rarely described. An example is the UniProt database [9] which provides an XML schema for the mapping of UniProt's XML format onto hierarchical structures of XML databases.

Informal descriptions allow to develop parsers manually by interpreting the given rules, but they are not suitable to generate parsers automatically. For automatic parser generation, however, a formal format description is indispensable. Formal descriptions enable machine processing. Examples for appropriate notations from computer science and bioinformatics are the Document Type Definition (DTD) for XML or the Abstract Syntax Notation One (ASN.1). ASN.1 is, for example, used at the National Center for Biotechnology Information (NCBI) for the specification of data types. The UniProt consortium uses XML/DTD to format flat files, e.g. the data exchange format of the UniProt database.

Especially for molecular biological databases, XML plays an important role in data formatting. The following list contains several XML-based data formats [1]:

- *Biopolymer Markup Language (BioML) [31]:* BioML was developed for modelling the hierarchical structures of organisms.
- *Chemical Markup Language (CML) [75]:* CML aims at managing different chemical information in connection with additional information, e.g. publications.

- *KEGG Markup Language (KGML)*[5]*:* KGML contains a DTD for the representation of metabolic pathways including metabolites and enzymes.
- *Systems Biology Markup Language (SBML) [47]:* SBML is a markup language for the representation of computational models in biology. It contains structures for describing subcellular loci (compartments), biochemical reactions and chemical entities involved. Parameters can be declared both globally (for all reactions) and locally (for a single reaction only). Furthermore, units and mathematical rules can be specified.
- *Taxonomic Markup Language [34]:* The Taxonomic Markup Language contains a DTD for the description of taxonomic relationships between organisms.

Apart from the above mentioned, many more XML-based data formats exist, e.g. CellML (Cell Markup Language) [20] or MAGE-ML (MicroArray and Gene Expression Markup Elements).[6] The ongoing development of standard formats for model representation is internationally being coordinated by the COMBINE initiative.[7]

3.2.4 Metadata

Not only business companies are losing hundreds of billions of US dollars per year due to bad *data quality* [27], this also holds true for other areas, including the research sector. For a meaningful use of data – not only in running projects, but also beyond – a high data quality is indispensable. Reaching this aim can be supported by the substantial use of *metadata*. Metadata is additional information provided together with the generated data. One major advantage of the availability of metadata is that they help to perform promising data analysis using data from different life science domains. Metadata is (structured) data describing a resource, an entity, an object or other data. It is used to retrieve, use and maintain a resource, an entity, etc. Unfortunately, often the acquisition of (primary) data and its subsequent processing are not well documented. For example, additional information, such as genotype, development and growth conditions, environmental conditions, tissue or treatment of biological objects, is missing at all or is described using different vocabularies. Further relevant information includes statistical methods or software tools and the parameters applied onto the data. Frequently, this lack of metadata leads to extra costs or additional personnel expenditures when aiming to reuse data or reproduce a result, e.g. when being forced to perform the same experiment multiple times.

[5] http://www.kegg.jp/kegg/xml/

[6] http://www.mged.org/Workgroups/MAGE/

[7] The computational modelling in biology network, COMBINE, http://co.mbine.org/.

The problems described above can be downsized by a complete and well-structured documentation of all steps starting with the acquisition of raw data and ending with the publication of results. Thus, the annotation of data with metadata is one important factor for its interpretation, reusability and structuring. This is reflected by manifold metadata schemata that are used in life sciences, mostly under the umbrella of the Minimum Reporting Guidelines for Biological and Biomedical Investigations (MIBBI) project [99]. Reporting guidelines define the minimum information necessary to be provided with a biological or biomedical experiment. The textual description of these information guidelines is often complemented by a data format encoding exactly that information in XML format (see Sect. 3.2.3) and providing mechanisms to link these XML elements with metadata in external resources, such as bio-ontologies, or technical information (e.g. file creators or modification dates for files). In general, it can be subdivided into *semantic* or *technical* metadata.

3.2.4.1 Semantic Metadata

Semantic metadata is closely connected to the scientific data domains and comprises an own universe of several hundreds of metadata schemata. For instance, in systems biology, a review summarises 30 different standards for metadata and data exchange formats [14]. *Ontologies* belong to semantic metadata. In computer science, an ontology is a definition of classes (concepts, objects) and their relationships (attributes, roles) [40]. It is well defined and contains the vocabulary of a data domain, thus improving the interoperability between systems or the communication between human beings.

Due to the growing amount of data in life sciences, it gets more and more important to put this data into relation. Therefore, ontologies are increasingly used [10]. Examples for life science ontologies are:

- *Gene Ontology (GO) [6]:* Molecular functions, biological processes and cellular components
- *Trait Ontology (TO) [50]:* Phenotypic traits of plants
- *Plant Ontology (PO) [7]:* Anatomy and developmental stages of plants
- *MGED Ontology (MO) [105]:* Annotation of microarray experiments

The BioPortal [106] maintains and integrates bio-ontologies that adhere to the requirements of the OBO foundry for open biological, high-quality ontologies [96]. Ontologies in the BioPortal can be browsed visually, and they contain cross-links to other OBO ontologies, enabling extensive exploration of biological knowledge, as well as thorough annotation of data. An *annotation* is a piece of meta-information accompanying a data set. It describes or explains the subject or content it refers to.

3.2.4.2 Technical and Administrative Metadata

Technical and administrative metadata cover aspects of management and processing of digital scientific resources. The collection and storage of structured technical metadata is an important prerequisite for the automatic management and processing of life science data sets. Technical metadata comprise aspects of how to access files, i.e. information about the system requirements for use in terms of hardware and software as well as the unique identification and documentation of the file format in which the resource exists. Each data set should have a unique, persistent identifier, which is identified regardless of its location.

For example, in life sciences, there is a deficiency of generally accepted conventions for referencing data records. Proprietary identifiers, such as so-called accession numbers, are designed as a unique combination of alphanumeric characters. For example, the proprietary identifier `Q8W413` in the UniProt database [69] refers to the protein `beta-fructofuranosidase`.[8] The enzyme `3.2.1.26`[9] points to the same entry but is interpreted as standard nomenclature for enzymes. In The Arabidopsis Information Resource (TAIR), the locus tag `At2g36190`[10] is an identifier for the coding gene of the same protein in the plant *Arabidopsis thaliana* (prefix *At*). Furthermore, the gene synonym `AtFruct6` is an example for a semantically enriched acronym of a gene: *At* denotes *Arabidopsis thaliana* and *Fruct* `beta-fructofuranosidase`.

To overcome this problem, tools have been designed that resolve identifiers and approaches to standardise technical metadata. Known resolver systems are, for example, identifiers.org [51] and the UniProt database identifier mapping.[11] Popular schemata for technical metadata are the Dublin Core Metadata Element Set (DCMES),[12] accepted as ISO standard 15836, as well as the closely related DataCite Metadata Schema.[13] DCMES was developed by scientists and librarians to homogeneously describe digital objects using 15 mandatory elements. The DataCite schema is less strict and comprises only 5 mandatory and 12 optional elements. However, the most popular way of primary data annotation remains to be semantically enriched file names.

[8] http://www.uniprot.org/uniprot/Q8W413

[9] http://www.expasy.org/enzyme/3.2.1.26

[10] http://www.arabidopsis.org/servlets/TairObject?type=locus&name=AT2G36190

[11] http://www.uniprot.org/?tab=mapping

[12] http://dublincore.org/documents/dces

[13] http://schema.datacite.org/meta/kernel-2.2/index.html

3.2.5 Database Integration

In general, *data integration* is a service combining contents of multiple, often heterogeneous, data sources, thus enabling to gain new insights [107]. In contrast to the integration of information systems in business companies, data integration in life sciences mainly focuses on combining data of *heterogeneous sources*, e.g. from the World Wide Web. According to [87], heterogeneity can be classified as (i) heterogeneity on systems level (different system properties of the sources, e.g. optimiser strategies), (ii) heterogeneity on data model level (use of different database models, e.g. relational or object-oriented model), (iii) heterogeneity on schema level (e.g. different representation of similar data) and (iv) heterogeneity on data level (e.g. different data for similar database objects).

Research in life sciences typically distinguishes two integration approaches [21]:

1. *Virtual (or logical) data integration:*
   This type of integration is often used for web-based data sources. Here, an integration system sends a query to several data sources and combines the results into a report at runtime. Since no data is stored locally, the results are always up to date, but the query performance is usually lower than with the materialised integration.
2. *Materialised (or physical) data integration:*
   Following this approach, data sources are queried for new data at regular intervals, and this data is stored locally. The integration system then queries the local data only, which has a higher performance than querying distributed sources as with the virtual integration. However, the timeliness of the locally stored data depends on the update intervals.

In the recent past, typical approaches using the virtual integration were *multi-database systems (MDBS)* and *mediator-based systems*. Multi-database systems extract data from several separate database systems and present this data using a homogeneous view [83]. In contrast to these systems, which focus on data stored in database systems, mediator-based systems [108] aim at integrating data stored outside of databases, e.g. HTML or flat files. The latter approach is widely used in bioinformatics. Examples for virtual integration in life sciences are Entrez [90], the Sequence Retrieval System (SRS) [30] and the Distributed Annotation System [26].

The typical approach using materialised integration is the *data warehouse (DWH)* approach which gained popularity in the end of the 1980s [23]. In contrast to *OnLine Transactional Processing (OLTP)* systems, which are designed for management of operative data (no historical data), data warehouses aim at providing non-volatile, aggregated and time-dependent data for analyse purposes, e.g. decision support. For setting up a data warehouse, data from different sources is extracted into a so-called staging area, transformed and then integrated into the data warehouse. *Data marts* are department-specific or application-specific and complement DWH, aiming at answering particular questions. Here, the two contradictory approaches of Inmon [49] (top-down approach) and Kimball [54] (bottom-up approach)

are distinguished. According to Inmon, all necessary data is stored in the data warehouse. Data marts are then derived from the data warehouse. In contrast, Kimball regards the creation of data marts as the beginning of the warehousing process. Thus, the data warehouse is a virtual collection of all data marts. Examples for materialised integration in life sciences are Atlas [92], BioMart [53, 95] or BioWarehouse [63].

The need for data integration in life sciences is increasing continuously [36]. So far, the aim of data integration was to provide a homogeneous view onto the integrated data. Recently, a paradigm change can be observed. As described in [19], it gets more and more accepted that different users need different kinds of data integration, because the semantics of data depends on its context. This change in thinking grounds in the fact that the number of scientific questions asked on the available data increased tremendously (e.g. due to high-throughput technologies). Consequently, extended possibilities of retrieving relevant information are necessary.

3.3 Information Retrieval

The increasing popularity of information retrieval as a method to handle semi-structured data and to formulate fuzzy queries correlates with the growth of data that is available online. This development is also reflected in milestones such as the triumphant throughout of PubMed as the world's most important biomedical literature search engine since 1996 [100].

Because of heterogeneity in both, the schema and the system, it is hardly possible to use structured query languages, i.e. SQL or OQL, to access the above-mentioned distributed data. In contrast, the tendency is to apply search engines or information systems to acquire speedily and precisely the information needed [24, 60, 68]. This promising technology is effective for knowledge and data published in journal articles or in its condensed form as hundreds of life science databases [32, 38].

Search engine technology provides efficient and intuitive IR methods to find relevant data in a collection of distributed, heterogeneously structured and modelled data repositories. *Desktop search engines*[14] like Windows Search or Strigi are popular at the scientists' desktops. Frameworks like Apache Solr[15] allow to embed full text search and relevance ranking into data repositories, as well as faceted search. The increasing availability and performance of this technology support the trend to replace classic query forms and Boolean query languages by keyword-based search and relevance filtering. This replacement gets increasingly important in life science information systems and is also implemented in primary data repositories, e.g. the DataCite Metadata Search.

[14] http://en.wikipedia.org/wiki/List_of_search_engines#Desktop_search_engines

[15] http://lucene.apache.org/solr/

Instead of referring to relevance ranking, in the following, the term *ranked retrieval* will be used, which expresses the necessity to provide an order for results from a data retrieval process. The interpretation of the term *order* is one central concept of ranked retrieval. Mathematically, it is a partially ordered set R, where R includes the result of a data retrieval query. Furthermore, for R a binary relation $<$ indicates that, for certain pairs of elements in the set, one of the elements precedes the other. In the context of ranked retrieval, the relation $r_1 < r_2 \mid r_1, r_2 \in R$ may have different definitions. The definition of this order relation is the focus of the ranking.

The order of query results becomes particularly important when a query comprises a high number of results. The user should have the possibility to structure and filter data, which are usually displayed as list of data records. If the data records comprise many fields with a high number of individual values, the result listing comprises data excerpts or even a list of access numbers, i.e. IDs. In that case, it is of particular importance to provide a useful order.

Empirically, the word "useful" could have very different meanings. This meaning is hardly dependent on the user's *pertinence*. There are cases when the order is defined by ordinal numbers, like publication date or serial numbers. Another order criteria is the lexicographic order. But numeric or lexicographic ordering is not necessarily a sufficient ranking criterion. Thus, defining relevance functions to determine the relevance of a data item and mapping it to an orderable p-value is one of the major challenges in IR.

In the following sections, two major categories of relevance ranking in life sciences will be discussed. The first category is the *explorative information retrieval* with the focus on an explorative and unbiased retrieval of data over a maximum set of databases, where the relevance ranking is mainly based on popularity and structure in the data itself. The second category, *semantic information retrieval*, is based on the presence of a model in a predefined network of data records that matches best to a very focused query. The model uses word associations and property lists.

3.3.1 Explorative Information Retrieval

Explorative information retrieval is a concept which bases on the idea of exploratory search [70] and represents the activities performed by researchers who are either:

- Unfamiliar with the domain of their goal
- Unsure about the ways to achieve their goals or
- Even unsure about their goals in the first place

In Fig. 3.3, the three major types of search are summarised as *lookup, learn, investigate* and classified into the activities *lookup search* and *exploratory search*.

Following this argumentation, explorative IR combines diverse methods of information retrieval, i.e. domain-specific text indexing, relevance feedback, relevance prediction or recommender systems, with human-computer interaction (HCI) in order to help users exploring data rather than performing lookup searches.

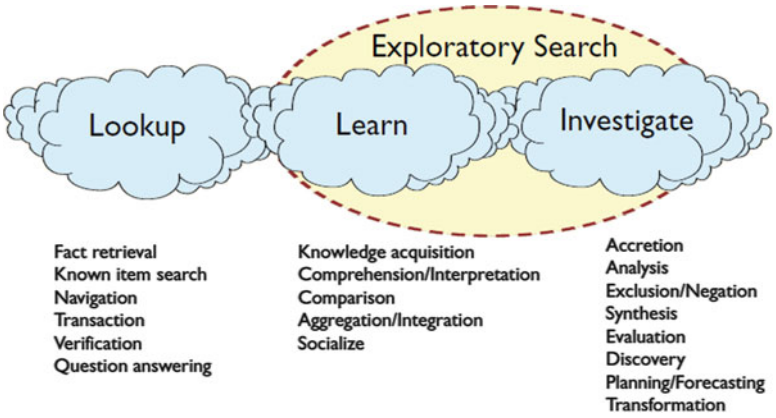

Fig. 3.3 Common search activities in web search, which are labelled as lookup–learn–investigate in [70] (©2006 Association for Computing Machinery, Inc. Reprinted by permission)

An up-to-date overview about the research activities on explorative IR can be found at http://en.wikipedia.org/wiki/Exploratory_search. Studies as the one described in Marti Hearst's book on Search User Interfaces [43] show that search behaviour evolves over time and is strongly influenced by the presence and capabilities of search engines. The main search engine experience of users is still contact with relevance-ranked search. To our experience, current prevalent strategy in bio information retrieval is ranked or Boolean search, combined with metadata-driven browsing and recommendation for exploration of data sets. However, new types of interfaces that emphasise exploratory search are also up-and-coming.

3.3.1.1 Relevance Ranking

"Just head for Google or Entrez and get the related web page or database entry." This is being said among biologists who search information about a certain object [24]. However, issues like finding reliable information about the function of a protein, or identifying the protein that is involved in a certain activity of the cell cycle, are much more challenging tasks. One has to choose (or screen) more than 1,512 life science databases and billions of database records [32].

Intuitively, the first choice for information acquisition are web search engines. Web site ranking techniques order query hits by relevance. However, trying to apply ranking methods that were developed to rank natural language text or WWW sites to life science content and databases is questionable [81]. For example, the top-ranked Google hit for `arginase` is a Wikipedia page. This is because the page is referenced by a high number of web pages or Google assigned a manual defined priority rank. Here, the hypothesis is: *A high hyperlink in-degree of a page means high popularity and high popularity means high relevance* [61].

In order to find scientifically relevant database entries, scientists need strong scientific evidence in relation to the specific research field. A dentist has other relevance criteria than a plant biologist or a patent agent. The intuitive and commonly used way at the scientist's desktop is query refinement. Criteria like who published in which journal, for which organism, evidence scores and surrounding keywords are of major importance. Even complete search guides are published, e.g. for dentists [22].

Other ranking algorithms use *term frequency – inverse document frequency (TF–IDF)* as ranking criteria. Apache Lucene[16] is a popular implementation of this concept and is frequently used in bioinformatics, like LuceGene from the GMOD project [77], which is used for the EBI search frontend EB-eye. The TF-IDF approach works well but misses the semantic context between the database entries and the query.

Another approach is probabilistic relevance ranking [48], where probabilistic values for the relevance of database fields and word combinations have to be predefined. In combination with a user feedback system, the probabilistic approach shows promising ranking performance [4].

Semantic search engines use methods from natural language processing, semantic tagging and dictionaries to predict the semantically most similar database entries. Such conceptual search strategies, implemented in GoPubMed [25] or ProMiner [41], are frequently used algorithms in text mining projects.

After choosing a ranking algorithm for a search engine, the next task is to define possible ranking criteria. Conventional search engines use several ranking criteria. Andrade and Silva consider the similarity between the result entry and the search query itself as a top-ranking criterion [5]. The importance of linkage in ranking has been put forward by PageRank, its variations and ranking extensions [81], which now constitute a mature field.

Greifeneder [39] proposes several possible relevance criteria, including the absolute or relative frequencies of the keyword(s) of the search query, the scope or the actuality of the web page constituting the query result.

Schöch also mentioned the shortness of a URL and the order and the proximity of the search query terms as a criterion [88]. Both Greifeneder and Schöch suggest to check the entries for their popularity [39, 88]. This idea is based on centrality computation, which is an important research area in network analysis. One popular example for this usage is the PageRank algorithm of Google [15, 61].

3.3.1.2 Recommender Systems

In its most common formulation, the recommendation problem is reduced to the problem of estimating ratings for the items that have not been seen by a user and would be of interest. Intuitively, this estimation is usually based on the ratings given

[16] http://lucene.apache.org

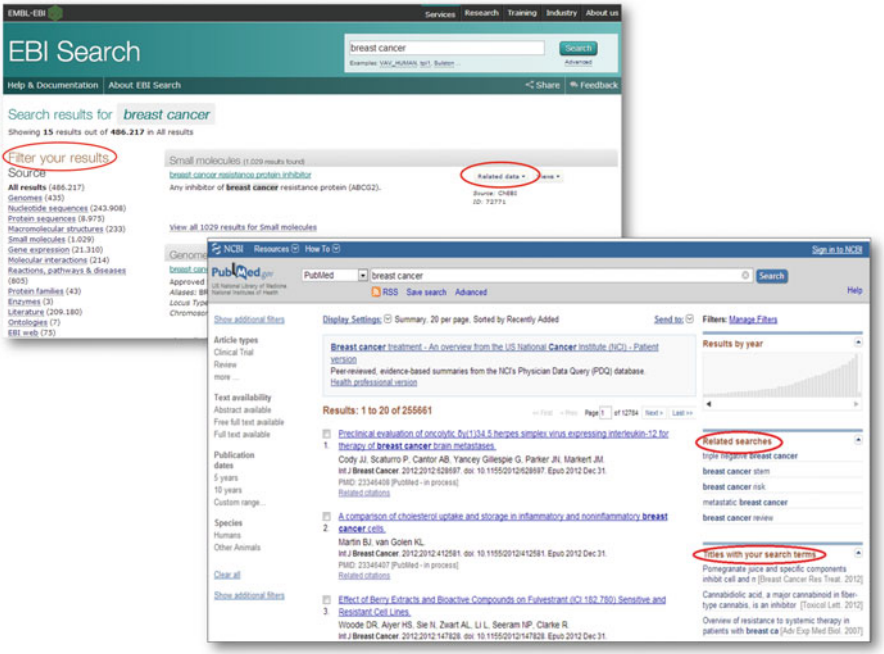

Fig. 3.4 Recommender systems used in the EBI's EB-eye IR system [37] (*left*) and NCBI PubMed literature search (*right*) – cross database search data or abstracts for the term "breast cancer" result in more than 486,000 hits in EBI databases and more than 255,000 in PubMed abstracts. The queries were executed at 2013/01/25. In PubMed, "Related searches" and "Titles with your search terms" suggest references using collaborative filtering. EB-eye makes intensive use of facets, which may be applied to incrementally refine the query and related documents using vector space model

by this user to other items and on some other information [3]. In *recommender systems*, the utility of a data record is usually represented by a rating, which indicates how a particular user liked a particular data set. An example of a user-item rating is PubMed's "Related searches" and "Titles with your search terms" (see Fig. 3.4).

Recommendation in life science IR can be divided into the phases *query expansion* and *related documents prediction*.

The first phase is *query expansion*. It describes the process of adding terms to or deleting terms from the original query. Here, a recommender system should anticipate from users strategies to find a pearl – the *citation pearl growing strategy* and the *building blocks strategy* [28]. In case of the building blocks strategy, the user divides the information retrieval problem into different concepts and assigns one or more reference terms to each concept. This is embedded into an incremental process of refinements until the most relevant document is selected by the user as local optimum. The *citation pearl growing strategy* uses intermediate query result, which is retrieved by a broad query, and interactively pick terms to expand the original query. The concepts can be implemented in automatic query expansion systems which make use of thesauri, ontologies and synonym lists and, in the case of pearl

picking, use top-ranked query results, for example, by collaborative user rating, and pick the relatively most frequent terms in the top documents to expand the query. An add-on is the syntactic expansion of single terms. This is done by computing edit distances to words in a dictionary, phonetic or word stem expansions. A popular implementation of these concepts is the facets. EBI's EB-eye IR system [37] and the information retrieval portal GoPubMed [25], which use the Gene Ontology [6] as thesaurus, are examples of successful application of facets in bioinformatics. Section 3.3.1.3 include some more elaborations to HCI, in particular facets.

The second phase is *related documents prediction* (also known as "more like this" or "page like this"). Based on a query result with relevance-ordered database records, the task of the recommender system now is to extend the result set with related documents. These related documents are not necessarily part of the core result set. There are five major methods proposed to predict such neighbour documents:

1. *Shared terminology:* Significant number of shared words; distance scoring using vector space model.
2. *Part-of data cluster:* Data records are part of the same data partition, i.e. synthetic genes and same species.
3. *Cross references:* Identifiers or explicit hyperlinks build data networks; distance scoring is used to predict neighbours [74].
4. *Collaborative filtering:* Follow users, who already (successfully) refined queries; filter user by client clustering, i.e. origin domain, country and user profile.
5. *Content-based recommendation:* Suggest data records, which were selected in past in a close query session/time context.

The above methods are rarely implemented in life science IR systems. Some of them apply shared terminology, cross references and part-of clusters, e.g. PubMed or EB-eye.

3.3.1.3 Human–Computer Interfaces

Marti Hearst gives in her book a literature-based overview about challenges in information retrieval interface design [43]. One interesting observation that she makes and that is easily verified is that even after 15 years of HCI in web search, general-purpose web search interfaces are still based on a one-line entry of search terms coupled with some query suggestions.

However, in the past 10 years, a new search paradigm emerged, called *Hierarchical Faceted Search (HFS)* [42]. This search paradigm is especially convincing for small, hand-picked data sets, i.e. the classic Nobel Prize Winners example available.[17] However, it has shown viability also for huge data sets such as search results in online stores.

[17] http://flamenco.berkeley.edu/demos.html

The goal of HFS is to enable users to explore data sets. It does so by guiding the user, as well as efficiently communicating progress of the search and a position within the collection. HFS is an improvement on classical hierarchical search. How this works can simply be illustrated using a search for car by brand, size class, and engine type. Each car has a given brand, a size class and engine type. They are facets describing the car.

Classifying a given set of cars into one hierarchy, one would have to choose which facet to put first. For example, should be browsed by engine type or rather by size class first? Once the hierarchy is chosen, every user will have to go down the predefined path to browse the cars collection.

The base innovation in HFS is to avoid this decision; instead it is accepted that each item in the cars collection has multiple facets. Each facet corresponds to a hierarchy of subsets, and each car is member of one subset for each of its facets. The faceted search interface enables the user to choose the important facets and to choose to which subsets a query result has to belong at the same time. For example, users want a small car, they do not care about the engine type and it must be a Chevrolet. They thus picked one subset of the size facet and one for the brand facet.

To get a feeling of the amazingly simple and intuitive browsing that can be achieved this way, try the flamenco Nobel Prize Winners demo. Please note how details play a big role in faceted search, for example, the display of query result sizes before the query in order to give a preview of what can be expected when clicking on a given facet.

While this example shows the advantages of faceted search, there are some inconveniences that keep faceted search from wider use for large data collections:

- Too many facets and too large fan-out of facet hierarchies: In free data collections, there is a huge amount of potential facets. It is impossible to show all of them on a screen.
- Absence of high-quality facet hierarchies: Annotated by hand, one can design high-quality facets; however, automatic classification in high-quality facets is hard.

GoPubMed (see example at Fig. 3.5) exemplifies strengths and challenges of faceted search for biologists: On the one hand, the interface enables browsing via facets, using the well-developed taxonomies that biology has to offer; on the other hand, browsing uses a lot of its intuitivity with the huge fan-out of bio-ontologies. GoPubMed counters this via emphasised display of *top concepts* and the possibility for logged-in users to define *favourite terms*. Other possibilities of countering the fan-out problem are subject of ongoing research. However, some systems recently started to include elements of faceted search in addition to classic search, e.g. the "browse targets" functionality in ChEMBL,[18] or autocompletion with display of result size previews in SABIO-RK.[19]

[18] https://www.ebi.ac.uk/chembl/malaria/target/browser/classification

[19] http://sabio.h-its.org

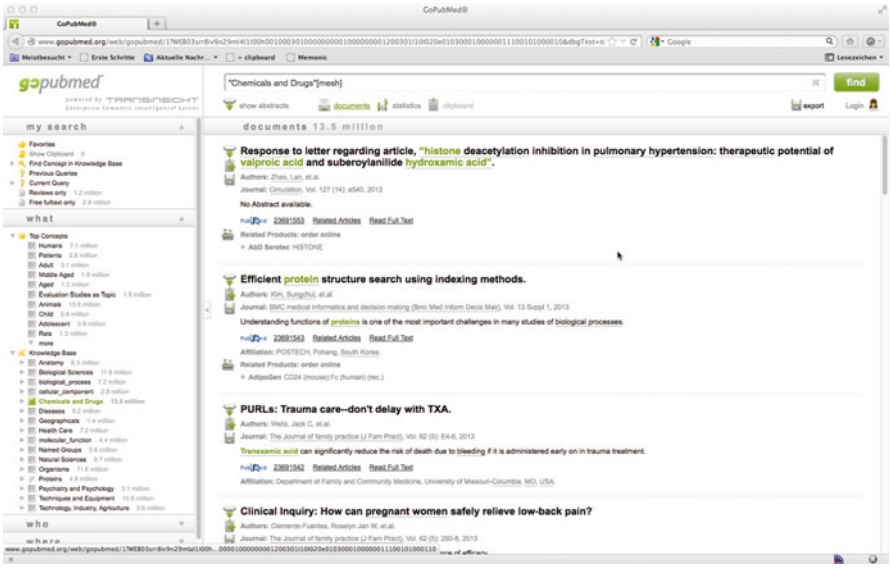

Fig. 3.5 GoPubMed example search. Notice how care is taken to limit the fan-out of trees, keeping it down to only 20 children of the "Knowledge Base" tree. However, already 20 entries have to be read one by one. Logged-in users could counter this by using bookmarked terms for future searches, thus creating search trails

3.3.1.4 The Explorative IR System LAILAPS

LAILAPS stands for "*L*ife Science *A*pplication for *I*nformation Retrieval and *L*ightweight *A*PI for *P*ortable *S*earch Engines" and as metaphor for the Greek mythological dog who never failed to catch the prey what he was hunting. In IR semantics, the aim is to provide a tool that supports the information discovery in the world's life science databases. This bold goal must meet continuously changing requirements. Some are gained from over 10 years experience in dozens of data management, database integration and analysis projects. The result is the development of the LAILAPS IR system. This project has been running for 6 years and combines state-of-the-art methods and concepts from the computer sciences, life sciences and bioinformatics. Empirically collected user requirements from bioinformaticians, IT-skilled biologists as well as less experienced students are used to design an intuitive user interface and feedback system. The first LAILAPS version was released in 2007 as an project that was coordinated by an European plant science company. Motivated by insufficient relevance ranking and the high number of unsorted query results from database query systems, the aim was to implement a search engine for protein databases with a user-specific relevance ranking model.

The approach was to import major public protein databases – i.e. UniProt, PIR and KEGG – into an in an EAV schema, decompose and tokenise the text,

Table 3.1 LAILAPS feature set to score database entries

Feature class	Description
Attribute	Attribute in which the query term was found
Database	Database origin of the entry
Frequency	Frequency of all query terms in the entry and attribute
Co-occurrence	Expresses how close and in which order the query term were found
Keyword	Rating of keyword semantics sorrounding the query hits
Organism	Organism to which the entry relates to
Raw data length	Length of the raw data, which is embedded in the database entry
Text position	Portion of the attribute covered by the query term
Synonym	Information if the hit was produced by an automatic synonym expansion

compute a reverse text index and compute scores for data entities. The concept of the LAILAPS query system is to support lists of search terms and phrases. A search result is a relevance-ranked list of database entries. Each entry is displayed in form of an rich snipped that summarised the content in one text line. The basis of the relevance ranking is a set of nine classes of features, which are shown in Table 3.1. The quantification of these features is computed for each result record as static entry properties or as from the properties of the text index search itself. The parameterisation of the relevance prediction algorithm is based on user feedback. The user may explicitly rate the page quality or the web browser tracks the user actions and estimates the page quality. This reference data is used to train user-specific neural networks, which predict from feature scores the page relevance. The initial training has been performed with a set of 1,089 manually relevance-rated protein entries that results from 19 queries [60]. A 80/20 cross validation shows a precision between 0.62 and 0.81, a recall of 1.00 and an f-score between 0.76 and 0.90.

The screenshots in Fig. 3.6 display the major components of the LAILAPS web application. A portlet version is available to embed LAILAPS into a custom web page.

Since 2011, the LAILAPS development is focused to support the explorative IR in a genomic context. Here, LAILAPS is used to bridge genomic metadata, like functional annotation to genes or other regions at a genome. The concept is:

1. Compile a domain specific list of data hubs, which acts as information retrieval core.
2. Text search and relevance ranking.
3. Reverse identifier lookup.

The implementation of this concept for the genomic data domain underlines the flexibility of LAILAPS concept. Here, the world's major resources of genomic data annotation are compiled in a list of eight major databases: Trait Ontology, Pfam, Gramene, Plant Ontology, SwissProt, TrEMBL, Gene Ontology and PDB. Those are indexed and linked back to the genomics data, i.e. the Genebank Informa-

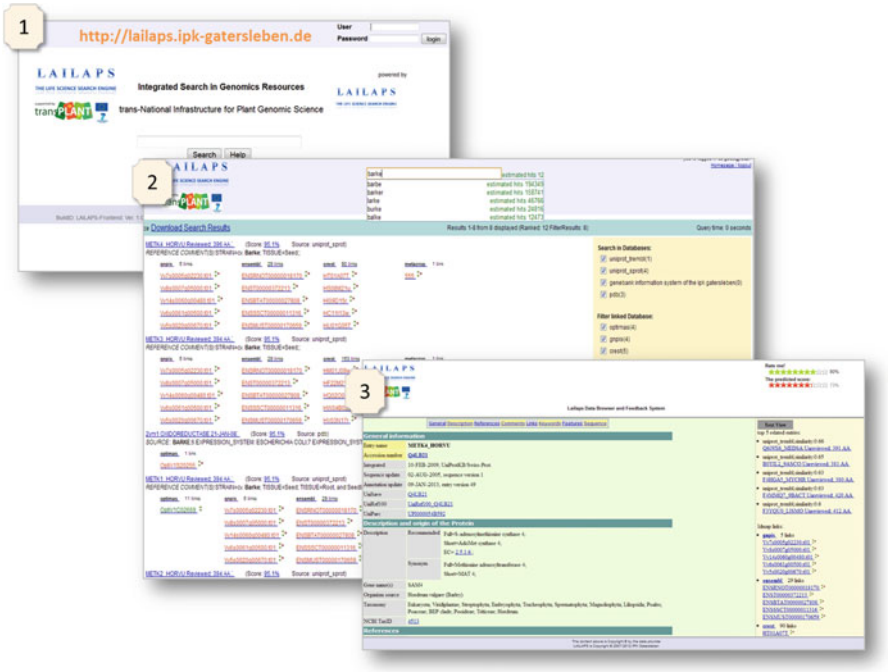

Fig. 3.6 The LAILAPS search engine for integrated search in transPLANT genomics data network. Part (*1*) shows the entry point of the search engine. In screenshot (*2*), a result of a keyword search for "barke", a genotype of barley, is shown. The result contains relevance-ranked hits in indexed genome annotation data hubs (UniProt, Gene Ontology, PFAM, etc.) and related linked genomic resources, i.e. Ensembl, GnpIS, CR-EST. In screenshot (*3*), the integrated data browser and feedback system, which act as input for the incremental training of the relevance predicting neural network

tion System (GBIS) of the German ex-situ Genbank,[20] EBI integrated genomics information system Ensembl,[21] and the INRA integrated genomics information system GNpIS[22] by the French INRA institute. The results of search queries are relevance-ordered links to genomic data. LAILAPS is part of the transPlant consortium to build a transnational plant genomic infrastructure and supported by the European Commission within its 7th Framework Programme, under the thematic area "Infrastructures". The implementation of this IR infrastructure is available at http://lailaps.ipk-gatersleben.de.

[20] http://gbis.ipk-gatersleben.de/gbis_i/home.jsf

[21] http://www.ensembl.org

[22] http://urgi.versailles.inra.fr/gnpis

3.3.2 Semantic Information Retrieval

The focus of this book chapter has so far been on the integration and retrieval of large-scale bioinformatics data. Another type of data that needs to be integrated are computational simulation models. During the past decades, modelling and simulation techniques have been used to answer biological questions. A consequence is the development of computational models, often in the area of systems biology. Systems biology is the study of complex biological systems by means of computational approaches and methods. A computational model of a biological system then represents aspects of that system, using, for example, mathematical equations. The number of available models has grown steadily over the last decade, and so has the models' complexity [44]. Models are being shared and reused in standard formats [102], so-called model representation formats (see Sect. 3.3.2.1). The increasing number of models is stored and managed in model repositories such as BioModels Database or PMR2 (see Sect. 3.3.2.2). To handle the models' increasing complexity, semantic annotation has been established as a tool to describe a model's nature. The novel research field of semantic systems biology investigates how to use these annotations to improve model management tasks such as model retrieval, model combination or version control. Section 3.3.2.3 focuses on annotation-based model retrieval and ranking.

3.3.2.1 Model Representation Formats and Standards

To reuse existing model code, the code itself must, first, be made available in model databases. Second, it must be encoded in exchangeable standard formats, which can then be interpreted by software tools. BioModels Database [66] is one example of an open model repository that freely distributes models in standard formats. Frequently used model representation formats are all XML based; examples are the aforementioned Systems Biology Markup Language (SBML [47]), CellML [20] or NeuroML [35] for models of neuroscientific investigations. These standard formats encode the necessary information to rebuild the model structure and underlying mechanisms in a software environment, e.g. for simulation studies.

Together with the model, a whole plethora of meta-information is provided, including the reference publication, the model authors, the semantics of the encoded entities, the model curation state, the underlying mathematics or the graphical representation of the model. Often, meta-information is encoded in bio-ontologies [12] (e.g. Gene Ontology, GO [6], the Systems Biology Ontology (SBO) [65] or the NCBI Taxonomy[23]) and linked to model entities through semantic annotations.

Model annotations mostly refer to technical and administrative information (see Sect. 3.2.4.2), while annotations of model components point to background

[23] http://www.ncbi.nlm.nih.gov/Taxonomy/

knowledge from biology or chemistry. The annotation information may either be contained in the model or it may be stored in an external file (see Sect. 3.2.4.1). As well as the model encoding itself, the annotation would best be provided in a standardised form, e.g. using the Resource Description Framework (RDF) [62]. RDF can be interpreted by a computer, and therefore RDF-encoded meta-information can automise tasks such as mode search, comparison, merging or clustering [44, 57, 91]. The ontology terms are in addition highly linked and therefore allow to infer further knowledge about the model.

Semantic annotations in RDF should follow the recommendation for model annotations, called MIRIAM guidelines [64]. The MIRIAM guidelines describe which additional information should be provided together with the model code and how it should be encoded. The SBML standard follows these recommendations and stores annotations as triplets of model entities, qualifiers and URIs pointing to an ontology entry (a so-called identifier [59]). For example, the XML element `species` represents an entity taking part in a biochemical reaction. The relation between the annotated XML element, e.g. the `species`, and the ontology reference, e.g. a GO identifier, is expressed also using standardised *qualifiers*.[24] The strongest relation is build up by the `IS` qualifier, i.e. the XML element *IS* exactly what is described in the ontology entry pointed to by the URI. Several weaker qualifiers exist, e.g. `isVersionOf`.

The meta-information encoded in model annotations is a major resource for information retrieval tasks. One prominent example is improved model search. For example, a user searching for models dealing with caffeine may express this search by typing `caffeine` or $C_8H_{10}N_4O_2$, or `1,3,7-trimethylpurine-2,6-dione`. A retrieval system is capable of finding the URIs pointing to ontology entries dealing with `caffeine` and relating them back to models that contain these URIs in their annotations. The basis is the creation of an index of terms from available ontology information. Researchers may use these terms, which best describe the nature of a particular molecule, to perform keyword-based searches. Keywords are more intuitive than cryptic model URIs or computer-generated entity names. If a model is properly annotated with ontology information about caffeine, then the IR-based search will also cover synonyms and external descriptions. Consequently, it is possible to retrieve models based on keywords that do not necessarily occur in the model code itself.

3.3.2.2 Exemplary Model Databases and Repositories

Models in exchangeable standard formats need also be stored and made publicly available to the modelling community to foster reuse. A number of databases and repositories have been established over the past years. The following is a brief review of selected model repositories [102].

[24] http://www.ebi.ac.uk/miriam/main/mdb?section=qualifiers

One distributor of freely available SBML models is BioModels Database [66]. To date it contains 436 curated and 497 non-curated models[25] and several thousands of automatically generated pathway models.[26] The majority of models in BioModels Database are concerned with signal transduction and metabolic processes. All models of the curated branch are guaranteed to be valid SBML and to reproduce the results described in the accompanying paper. Internally, metadata is extracted and stored in a MySQL database. Metadata includes information about the submission and modification dates of a model file, authors' information, references and annotations encoded as the aforementioned MIRIAM identifiers. Additionally, Apache Lucene is used to index a subset of model elements and metadata. BioModels Database supports browsing and searching for models. One way to browse is the list of available models (sorted by BioModels Database ID (BMID), model name, publication ID or date of last modification). Another way is to use a tree-structured browser that is based on GO terms. When searching for a model, a so-called multistep search is performed [66]. The system works in three sequential steps. Given a search term, first, the metadata, publications and the annotations stored in the MySQL database are queried. The result of this search is a set of BMIDs. Secondly, the stored SBML XML files are queried, using the previously generated indexes and parsing information such as the SBML `notes` tag. The returned BMIDs are added to the result set. If the search included query terms from external resources, then, thirdly, supplementary information is searched, using either information available in the local MySQL database or web services. For the specific case of searching for a term in a taxonomy, the taxonomy tree is also traversed for neighbour terms, and model IDs associated with that term are added to the result set. The output is generated by using the BMIDs to query the MySQL database for the formerly extracted metadata that is necessary for display on the web site. Search results are returned in an unordered result set.

The Physiome Model Repository (PMR2, [109]) is an online repository for CellML models at different stages of curation. The Plone-based Content Management System contains models of a wide range of different biological processes, including signal transduction pathways, metabolic pathways, electrophysiology, immunology, cell cycle, muscle contraction and mechanical models [67]. PMR2 intends to foster the processes of model curation and annotation so that ideally all models replicate the results in the published paper and the search for models and elements within models is facilitated. Models in the CellML Model Repository are browsed by different (physiological) categories, including cell cycle, signal transduction or metabolism. A CMS-wide full-text search allows for simple free text search. A search by particular model features (e.g. specifically by author or publication year) is not possible. Search results are returned in an unordered result set.

[25] Twenty-fourth release of BioModels Database, December 2012.

[26] http://code.google.com/p/path2models/

ModelDB [45] is a format-independent database for curated models related to computational neuroscience. It provides authors a repository for the storage of models, in particular in preparation for submission in neuroscience journals. ModelDB accepts models in any language for any environment [45]. It keeps the originally submitted model files, that is, the complete code specifying the attributes of the original biological system represented in the model, including interface and control code to run the model in the associated simulation environment, and a non-standardised readme text file explaining briefly how to use the provided computer code. Additionally, ModelDB stores model meta-information, including a concise statement of the model purpose, how to use it and a complete citation of the reference publication [45]. The underlying database management system is Oracle 10. as an instance of the Entity–Attribute–Value/Classes–Relationship framework (EAV/CR, [71]) for data representation. The search functionality in ModelDB relies on the meta-information entered by the model submitter. Search by author name or accession number (ModelDB ID) is supported. The complete list of models can be returned sorted by the model name or by the author. Additionally, some predefined queries regarding different criteria such as cell type or simulators are available. However, the queries do not incorporate the model files themselves; as such a search on the model code is not possible. The meta-information is not standardised, but consists of partially predefined strings and partially manually entered data. Third-party knowledge is not incorporated in the search process; the submitted models are not annotated.

JWS Online Model Database is part of the JWS Online Simulator [78], a web-based simulator for biochemical kinetic models. The model repository serves as the maintainer for a number of kinetic models that can be interactively run online. It supports the search for SBML models by a limited number of characteristics, including the author, publication title and journal, organism or model type. A web-based tool offers a searchable categorisation of models in the repository, distinguishing, for example, between cell cycle models and metabolism. A full-text search is not supported. Search results are returned ordered by author name. As there does not exist a publication on the technical background of the model repository, further information about the backend of the provided interface cannot be provided.

3.3.2.3 Model Retrieval and Ranking

A common shortcoming of all above mentioned model repositories is their limited ability to retrieve and rank models. A query containing domain-specific keywords retrieves an unordered set of models. Thus, it is up to the user to browse the results and inspect the models manually. The keywords searched for are not necessarily present in a model itself; however, they might be related to a model by an annotation. Progress in model search has been made with recently developed IR methods for ranked model retrieval [44]. We elucidate here how a keyword-based model search retrieves ranked results using the aforementioned model from

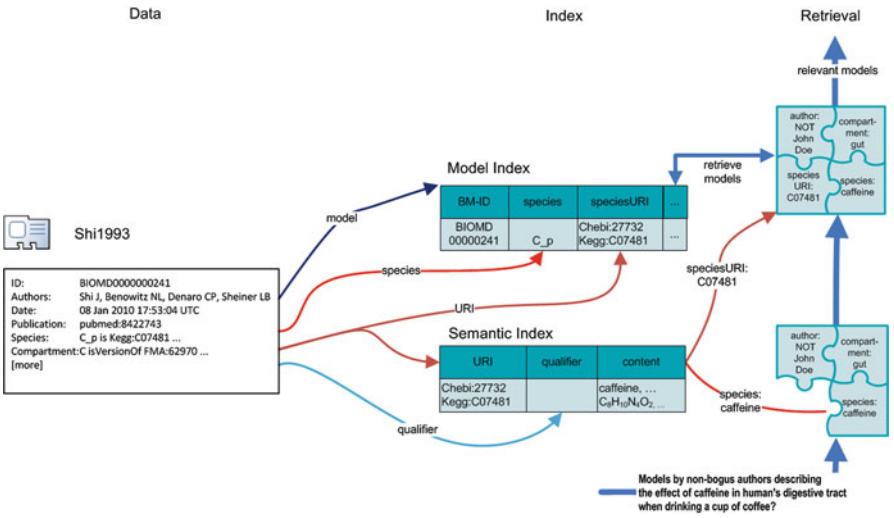

Fig. 3.7 This figure shows what model information is stored into the model and semantic index. Additionally, the search is expanded to retrieve models according to their biological content

BioModels Database.[27] This SBML-encoded model contains five compartments, five species, five rate rules and one assignment rule. Even though the model is all about `caffeine` (see example from Sect. 3.3.2.1), related keywords like $C_8H_{10}N_4O_2$, `1,3,7-trimethyl-3,7-dihydro-1H-purine-2,6-dione` or `guaranine` will not retrieve the model at all. This problem is solved by incorporating a model annotation. Figure 3.7 shows an excerpt of the example model. The model index holds information directly encoded in the model, i.e. the model's name, species or compartment names and also URIs used to annotate model entities. The semantic index in addition stores all URIs and links back to models. Here the textual content behind each URI is resolved and indexed.

The model retrieval is then performed using multiple steps. First, the specific query is sent to the model index. If no models or only models matching poorly on the query are retrieved, the search can be refined using the semantic index. Here, the keywords are used to identify matching URIs used to annotate models. As URIs link back to their corresponding models, it is possible to retrieve models using keywords not encoded in the model itself. Such a query expansion is shown in Fig. 3.7 where the term caffeine is used to add URIs to the original query. After all matching models are retrieved, a score is computed for each match. The score mostly relies on the concept of term frequency and inverse document frequency (see Sect. 3.3 for explanation). However, also the importance of certain model components is taken into account, e.g. a species is more important than a parameter value. In case of

[27] http://www.ebi.ac.uk/biomodels-main/BIOMD0000000241

URIs, also the relation between URI and annotated entity denoted by the qualifier is taken into account. A deeper explanation is given in [44]. The described approach can be tested on BioModels Database.

Additional possibilities for model search emerge if the networks spanned by several ontologies are integrated. Here, the so-called cross-links can be established and evaluated. One approach is the Bio2RDF[28] project which makes use of the vast information encoded in life science databases. The basic idea is to convert and to link the database contents with semantic web technologies [76]. After converting and linking, each database provides a SPARQL point [80]. The SPARQL point allows to create sophisticated queries on multiple data providers who also offer a SPARQL point. As a result, a number of RDF-triples matching the query are retrieved. Bio2RDF heavily uses semantic web technologies, allowing for automatic traversal through the network. An integrated network of ontologies can be used with OWL-based reasoning methods to identify model similarities (e.g. [46]).

In the ranked retrieval approach, which is closely related to a hierarchical faceted search from Sect. 3.3.1.3, the starting point when querying such a network of ontologies is one particular ontology entry, e.g. `xanthine`. If a user is interested in models revealing information about `xanthine` and its derivatives, a URI pointing to the `xanthine` entry is fed into the system. Thus, the descendants are retrieved and added up, along with inter-ontology links for the specific entry, to form a query. Finally, the query is sent to the model index, and a ranked list of models is retrieved.

3.4 Summary

Due to the increasing demands for data management in the life sciences, information retrieval is no longer just a buzzword. It has instead become a core concept in bioinformatics and related research fields. However, while project proposals still continue to ask for more storage in their budget plans, the aim should be to develop methods for more efficient use of storage. The mere drop of files to the largest possible secondary storage devices, i.e. hard drives or cloud storage solutions, could mean a dead end. Current practice is the storage of working files using a sophisticated naming system for files in combination with Microsoft Excel sheets to link some metadata. This is particularly true for many wet lab desks, and it may be suitable for personal- or even-group level data maintenance. The drawbacks of this system, however, become obvious in its publication process. Highly personalised data representation makes the data only discoverable by insiders, computer scientists or skilled bioinformaticians. The data of interest first needs to be transferred into well-modelled, granular structured and well-interfaced database systems before being reused. A main argument for data reuse is that the distribution of knowledge and later processing by computational analysis is essential to all scientific work.

[28] http://bio2rdf.org/

In order to meet the demands expressed above, this chapter gave an overview of core methods and technologies for modern information management in life sciences. The first focus was on databases and information systems. In this context, the change from flat file data exchange to relational database modelling over static database integration approaches to flexible data networks using semantic technologies has been described. Particularly exciting is the vision of a holistic view of a universe of thousands of single yet integrated, well-structured databases. This is, in fact, the real value of the data collected so far. It is not in the form of daily reinvented project-related scripts. The development of such scripts demands time and expert knowledge, and sometimes magic parameters and access paths are used. In contrast, reusable frameworks such as open templates for a workflow-driven data analysis should be preferred. The objective here is a sufficient standardisation and semantic enrichment of the data.

Obviously, the creation of reusable frameworks is a laborious and costly process. However, the overall gain for science will be even bigger. Therefore, lab staff needs to be motivated to use lab information systems and to maintain their protocols, observations and files in database systems. It continues at the scientist's level, where the data streams should be consolidated and properly semantically tagged, long-term citable stored and linked in a scientific publication as supplemental material, preferable in the already established domain databases. Finally, bioinformaticians should place emphasis on the code and interface quality. Besides coding, scripting and data analysis under time pressure, the potential lies in well-documented, object-oriented developed and well-tested software as well as in the use of standard data access protocols and interfaces. This enables the global scientific community to extract all possible knowledge from the existing data.

In addition to the granular and integrated access to globally distributed data, the selective access to information and their extraction is very important. Not the mere of data volume matters. The high number of, on the first view separated, but from a different perspective overlapping, data domains is often the most important cost factor for information retrieval. It could be argued that the actual core of the information retrieval is to find data and ultimately obtain information. This concern is mainly reflected in the section information retrieval. The section has been written with a focus on techniques and actual systems. Here, two most interesting aspects were described in summary – the exploratory and the semantic retrieval.

The focus of the first is on relevance ranking in a set of data query results and recommender systems to improve the query sensitivity and to filter the most important data items in respect to the user's needs. The second focus is on semantic information retrieval, such as the use of metadata or semantic networks and, finally, semantically interpreted data queries.

In this chapter, no evaluations of or recommendations for specific methods or systems were made. This is due to the fact that such evaluations strongly depend on actual applications, which are existing in a wide variety in life sciences. Instead, an extensive list of references of relevant sources in primary literature as well as of web sources was added, which should be seen as a starting point of own detailed studies of the readers.

Acknowledgements This work was supported by the European Commission within its 7th Framework Programme, under the thematic area "Infrastructures", contract number 283496, by the BMBF e:bio programme (University of Rostock) and the Leibniz Institute of Plant Genetics and Crop Plant Research (IPK).

WWW Link List

Resource	Brief description	WWW link
PubMed	PubMed comprises citations for biomedical literature	http://www.ncbi.nlm.nih.gov/pubmed
DBLP	The Computer Science Bibliography provides bibliographic information on major computer science journals and proceedings	http://dblp.uni-trier.de
SOAP	The Simple Object Access Protocol is a protocol specification for exchanging structured information in computer networks	http://www.w3.org/TR/soap
REST	Representational State Transfer is a style of software architecture for distributed systems such as the World Wide Web	http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm
KGML	KEGG Markup Language (KGML) is an exchange format of the KEGG pathway maps	http://www.kegg.jp/kegg/xml
MAGE	MicroArray and Gene Expression MAGE aims to provide a standard for the representation of microarray expression data	http://www.mged.org/Workgroups/MAGE
COMBINE	COMBINE (Computational Modeling in Biology Network) is an initiative to coordinate the development of the various community standards and formats for computational models	http://co.mbine.org
UniProt	UniProt provides a comprehensive, high-quality and freely accessible resource of protein sequence and functional information	http://www.uniprot.org/uniprot
ENZYME	The Enzyme nomenclature database (ENZYME) is a repository of information relative to the nomenclature of enzymes	http://www.expasy.org/enzyme
TAIR	The Arabidopsis Information Resource (TAIR) maintains a database of genetic and molecular biology data for the model plant Arabidopsis thaliana	http://www.arabidopsis.org
DCES	The Dublin Core Metadata Element Set (DCES) is a vocabulary of 15 properties for use in resource description	http://dublincore.org/documents/dces

(continued)

(continued)

Resource	Brief description	WWW link
MDS	The DataCite Metadata Store (MDS) is a service for data publishers to mint DOIs and register associated metadata	http://mds.datacite.org/
Wikipedia List of Search Engines	List of search engines, including web search engines, selection-based search engines, metasearch engines, desktop search tools and web portals and vertical market web sites that have a search facility for online databases	http://en.wikipedia.org/wiki/List_of_search_engines
Apache Solr	Solr™ is the popular, blazing fast open-source enterprise search platform from the Apache LuceneTM project	http://lucene.apache.org/solr
Explorative IR	Wikipedia overview about the research activities on explorative information retrieval	http://en.wikipedia.org/wiki/Exploratory_search
Apache Lucene	The Apache Lucene™ project develops open-source search software	http://lucene.apache.org
Flamenco	Flamenco search interface framework has the primary design goal of allowing users to move through large information spaces in a flexible manner	http://flamenco.berkeley.edu
Malaria Data Target Classification Hierarchy	Example of faceted search in Malaria Data in addition to classic search	https://www.ebi.ac.uk/chembl/malaria/target/browser/classification
SABIO-RK	SABIO-RK is a curated database that contains information about biochemical reactions and their kinetic rate equations with parameters and experimental conditions	http://sabio.h-its.org
LAILAPS	LAILAPS (Life Science Application for Information Retrieval and Lightweight API for Portable Search Engines) aims to support the information discovery in the world's life science databases	http://lailaps.ipk-gatersleben.de
Ensembl	The Ensembl project produces genome databases for vertebrates and other eukaryotic species and makes this information freely available online	http://www.ensembl.org
GBIS/I	Query portal to retrieve information from the German federal ex situ seed collection	http://gbis.ipk-gatersleben.de/gbis_i/home.jsf
GnPIS	Genetic and Genomic Information System is a tool aiming to provide simple and fast access to the data located in all URGI (plant and fungi data integration) databases	http://urgi.versailles.inra.fr/gnpis
NCBI Taxonomy	The Taxonomy Database is a curated classification and nomenclature for all of the organisms in the public sequence databases	http://www.ncbi.nlm.nih.gov/Taxonomy

(continued)

(continued)

Resource	Brief description	WWW link
BioModels.net qualifiers	The qualifier of an annotation should reflect the relationships between the biological objects represented by the model element and the annotation	http://biomodels.net/qualifiers
path2models	The purpose of the project is to systematically generate mathematical models corresponding to the entire KEGG pathways and submit them to BioModels Database	http://code.google.com/p/path2models
BioModels Database	BioModels Database is a repository hosting computational models of biological systems	http://www.ebi.ac.uk/biomodels-main
Bio2RDF	Integration of ontology networks into biomodel search	http://bio2rdf.org/
Identifiers.org	Identifiers.org is a system providing resolvable persistent URIs used to identify data	http://identifiers.org
SPARQL Query Language	SPARQL can be used to express queries across diverse data sources, whether the data is stored natively as RDF or viewed as RDF via middleware	http://www.w3.org/TR/rdf-sparql-query

References

1. Achard F, Vaysseix G, Barillot E (2001) XML, bioinformatics and data integration. Bioinformatics 17(2):115–125
2. Adams M, Kelley J, Gocayne J, Dubnick M, Polymeropoulos M, Xiao H, Merril C, Wu A, Olde B, Moreno R, Kerlavage A, McCombie W, Venter J (1991) Complementary DNA sequencing: expressed sequence tags and human genome project. Science 252(5013):1651–1656
3. Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans Knowl Data Eng 17(6):734–749
4. Agichtein E, Brill E, Dumais S (2006) Improving web search ranking by incorporating user behavior information. In: SIGIR'06: proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval, Seattle. ACM, New York, pp 19–26
5. Andrade L, Silva MJ (2006) Relevance ranking for geographic IR. In: Workshop on geographic information retrieval, SIGIR'06, Seattle
6. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, Harris MA, Hill DP, Issel-Tarver L, Kasarskis A, Lewis S, Matese JC, Richardson JE, Ringwald M, Rubin GM, Sherlock G (2000) Gene ontology: tool for the unification of biology. Nat Genet 25(1):25–29
7. Avraham S, Tung CW, Ilic K, Jaiswal P, Kellogg EA, McCouch S, Pujar A, Reiser L, Rhee SY, Sachs MM, Schaeffer M, Stein L, Stevens P, Vincent L, Zapata F, Ware D (2008) The plant ontology database: a community resource for plant structure and developmental stages controlled vocabulary and annotations. Nucl Acids Res 36(suppl_1):D449–D454
8. Baeza Yates RA, Neto BR (1999) Modern information retrieval. Addison-Wesley Longman Publishing Co., Inc., Boston
9. Bairoch A, Apweiler R, Wu CH, Barker WC, Boeckmann B, Ferro S, Gasteiger E, Huang H, Lopez R, Magrane M, Martin MJ, Natale DA, O'Donovan C, Redaschi N, Yeh LL (2005) The universal protein resource (UniProt). Nucl Acids Res 33(suppl_1):D154–D159

10. Bard JBL, Rhee SY (2004) Ontologies in biology: design, applications and future challenges. Nat Rev Genet 5(3):213–222
11. Bentley DR, Balasubramanian S, Swerdlow HP, Smith GP, Milton J, Brown CG, Hall KP, Evers DJ, Barnes CL, Bignell HR et al (2008) Accurate whole human genome sequencing using reversible terminator chemistry. Nature 456(7218):53–59
12. Bodenreider O, Stevens R (2006) Bio-ontologies: current trends and future directions. Brief Bioinform 7(3):256–274
13. Botstein D, White R, Skolnick M, Davis R (1980) Construction of a genetic linkage map in man using restriction fragment length polymorphisms. Am J Hum Genet 32(3):314–331
14. Brazma A, Krestyaninova M, Sarkans U (2006) Standards for systems biology. Nat Rev Genet 7:593–605
15. Brin S, Page L (1998) The anatomy of a large-scale hypertextual Web search engine. In: Proceedings of the seventh international conference on world wide web 7, Brisbane, vol 30. Elsevier, Amsterdam, pp 107–117
16. Brockschmidt K (1995) Inside OLE, 2nd edn. Microsoft Press, Redmond
17. Bry F, Kröger P (2003) A computational biology database digest: data, data analysis, and data management. Distrib Parallel Databases 13(1):7–42
18. Codd EF (1970) A relational model of data for large shared data banks. Commun ACM 13(6):377–387
19. Cohen-Boulakia S, Leser U (2011) Next generation data integration for life sciences. In: Proceedings of the 2011 IEEE 27th international conference on data engineering (ICDE'11), Hannover. IEEE Computer Society, Los Alamitos, pp 1366–1369
20. Cuellar A, Lloyd C, Nielsen P, Bullivant D, Nickerson D, Hunter P (2003) An overview of cellmL 1.1, a biological model description language. Simulation 79(12):740–747
21. Davidson S, Overton C, Buneman P (1995) Challenges in integrating biological data sources. J Comput Biol 2(4):557–572
22. Day J (2001) The quest for information: a guide to searching the internet. J Contemp Dent Pract 2(4):033–043
23. Devlin B, Murphy P (1988) An architecture for a business and information system. IBM Syst J 27(1):60–80
24. Divoli A, Hearst M, Wooldridge MA (2008) Evidence for showing gene/protein name suggestions in bioscience literature search interfaces. In: Pacific symposium on biocomputing, Kohala Coast, vol 13, pp 568–579
25. Doms A, Schroeder M (2005) GoPubMed: exploring PubMed with the Gene ontology. Nucl Acids Res 33(suppl_2):W783–W786
26. Dowell R, Jokerst R, Day A, Eddy S, Stein L (2001) The distributed annotation system. BMC Bioinform 2(1):7
27. Eckerson WW (2002) Data quality and the bottom line: achieving business success through a commitment to high quality data. TDWI report series, The Data Warehousing Institute, Seattle
28. Efthimiadis EN (2000) Interactive query expansion: a user-based evaluation in a relevance feedback environment. J Am Soc Inf Sci 51(11):989–1003
29. Elmasri R, Navathe SB (2000) Fundamentals of database systems, 3rd edn. Addison-Wesley, Reading
30. Etzold T, Harris H, Beaulah S (2003) SRS: an integration platform for databanks and analysis tools in bioinformatics. In: Lacroix Z, Critchlow T (eds) Bioinformatics: managing scientific data. Morgan Kaufmann, San Francisco, pp 109–145
31. Fenyö D (1999) The Biopolymer markup language. Bioinformatics 15(4):339–340
32. Fernández-Suárez XM, Galperin MY (2013) The 2013 nucleic acids research database issue and the online molecular biology database collection. Nucl Acids Res 41(D1):D1–D7
33. Geiger K (1995) Inside ODBC: [Der Entwicklerleitfaden zum Industriestandard für Datenbank-Schnittstellen]. Microsoft Press, Unterschleissheim
34. Gilmour R (2000) Taxonomic markup language: applying XML to systematic data. Bioinformatics 16(4):406–407

35. Gleeson P, Crook S, Cannon R, Hines M, Billings G, Farinella M, Morse T, Davison A, Ray S, Bhalla U et al (2010) Neuroml: a language for describing data driven models of neurons and networks with a high degree of biological detail. PLoS Comput Biol 6(6):e1000815
36. Goble C, Stevens R (2008) State of the nation in data integration for bioinformatics. J Biomed Inform 41(5):687–693
37. Goujon M, Valentin F, Miyar T, McWilliam H, Lopez R (2007) The EB-eye. EMBnetnews 13(4):18–21
38. Gray J (2007) Jim gray on eScience: a transformed scientific method. Retrieved from http://research.microsoft.com/en-us/collaboration/fourthparadigm/4th_paradigm_book_jim_gray_transcript.pdf
39. Greifeneder H (2010) Erfolgreiches SuchmaschinenMarketing: Wie Sie bei Google, Yahoo, MSN & Co. ganz nach oben kommen, 2nd edn. Gabler Verlag
40. Gruber TR (1993) A translation approach to portable ontology specifications. Knowl Acquis 5(2):199–220
41. Hanisch D, Fundel K, Mevissen HT, Zimmer R, Fluck J (2005) Prominer: rule-based protein and gene entity recognition. BMC Bioinform 6(Suppl_1):S14
42. Hearst M (2006) Design recommendations for hierarchical faceted search interfaces. In: ACM SIGIR workshop on faceted search, Seattle
43. Hearst M (2009) Search user interfaces. Cambridge University Press, Cambridge/New York
44. Henkel R, Endler L, Peters A, Le Novère N, Waltemath D (2010) Ranked retrieval of computational biology models. BMC Bioinform 11(1):423
45. Hines M, Morse T, Migliore M, Carnevale N, Shepherd G (2004) Modeldb: a database to support computational neuroscience. J Comput Neurosci 17(1):7–11
46. Hoehndorf R, Dumontier M, Gennari JH, Wimalaratne S, de Bono B, Cook DL, Gkoutos GV (2011) Integrating systems biology models and biomedical ontologies. BMC Syst Biol 5(1):124
47. Hucka M, Bergmann F, Keating S, Schaff J, Smith L (2010) The systems biology markup language (SBML): language specification for level 3 version. http://sbml.org/Documents/Specifications/SBML_Level_3/Version_1/Core
48. Ide NC, Loane RF, Demner-Fushman D (2007) Essie: a concept-based search engine for structured biomedical text. J Am Med Inform Assoc 14(3):253–263
49. Inmon W (2005) Building the data warehouse, 4th edn. Wiley, Indianapolis
50. Jaiswal1 P, Ware D, Ni J, Chang K, Zhao W, Schmidt S, Pan X, Clark K, Teytelman L, Cartinhour S, Stein L, McCouch S (2002) Gramene: development and integration of trait and gene ontologies for rice. Comparative and Functional Genomics 3(2):132–136
51. Juty N, Le Novère N, Laibe C (2012) Identifiers.org and miriam registry: community resources to provide persistent identification. Nucl Acids Res 40(D1):D580–D586
52. Kanz C, Aldebert P, Althorpe N, Baker W, Baldwin A, Bates K, Browne P, van den Broek A, Castro M, Cochrane G, Duggan K, Eberhardt R, Faruque N, Gamble J, Diez FG, Harte N, Kulikova T, Lin Q, Lombard V, Lopez R, Mancuso R, McHale M, Nardone F, Silventoinen V, Sobhany S, Stoehr P, Tuli MA, Tzouvara K, Vaughan R, Wu D, Zhu W, Apweiler R (2005) The EMBL nucleotide sequence database. Nucl Acids Res 33(suppl_1):D29–D33
53. Kasprzyk A (2011) Biomart: driving a paradigm change in biological data management. Database 2011:bar049
54. Kimball R (1998) Bringing up supermarts – a step-by-step approach to building a data warehouse from granular data. DBMS and Internet Syst 11(1):47–53
55. Kitano H (2002) Systems biology: a brief overview. Science 295:1662–1664
56. Krallinger M, Valencia A, Hirschman L (2008) Linking genes to literature: text mining, information extraction, and retrieval applications for biology. Genome Biol 9(Suppl 2):S8
57. Krause F, Uhlendorf J, Lubitz T, Schulz M, Klipp E, Liebermeister W (2010) Annotation and merging of SBML models with semanticsbml. Bioinformatics 26(3):421–422
58. Lacroix Z, Critchlow T (2003) Bioinformatics: managing scientific data. Morgan Kaufmann, San Francisco

59. Laibe C (2011) Identifiers. org and miriam registry: perennial identifiers for crossreferencing purposes. Available from Nature Precedings. http://dx.doi.org/10.1038/npre.2011.6479.1
60. Lange M, Spies K, Bargsten J, Haberhauer G, Klapperstück M, Leps M, Weinel C, Wünschiers R, Weißbach M, Stein J, Scholz U (2010) The LAILAPS search engine: relevance ranking in life science databases. J Integr Bioinform 7(2):e110
61. Langville AN, Meyer CD (2006) Google's PageRank and beyond: the science of search engine rankings. Princeton University Press, Princeton
62. Lassila O, Swick RR, Consortium WWW (1998) resource description framework (RDF) model and syntax specification. http://www.w3.org/1998/10/WD-rdf-syntax-19981008
63. Lee T, Pouliot Y, Wagner V, Gupta P, Stringer-Calvert D, Tenenbaum J, Karp P (2006) BioWarehouse: a bioinformatics database warehouse toolkit. BMC Bioinform 7(1):170
64. Le Novère N, Finney A, Hucka M, Bhalla U, Campagne F, Collado-Vides J, Crampin E, Halstead M, Klipp E, Mendes P et al (2005) Minimum information requested in the annotation of biochemical models (MIRIAM). Nat Biotechnol 23(12):1509–1515
65. Le Novère N, Courtot M, Laibe C (2006) Adding semantics in kinetics models of biochemical pathways. In: Proceedings of the 2nd international symposium on experimental standard conditions of enzyme characterizations, Ruedesheim
66. Li C, Donizelli M, Rodriguez N, Dharuri H, Endler L, Chelliah V, Li L, He E, Henry A, Stefan M et al (2010) Biomodels database: an enhanced, curated and annotated resource for published quantitative kinetic models. BMC Syst Biol 4(1):92
67. Lloyd C, Lawson J, Hunter P, Nielsen P (2008) The cellmL model repository. Bioinformatics 24(18):2122–2123
68. Lu Z (2011) PubMed and beyond: a survey of web tools for searching biomedical literature. Database 2011:baq036
69. Magrane M, UniProt Consortium (2011) UniProt Knowledgebase: a hub of integrated protein data. Database 2011:bar009
70. Marchionini G (2006) Exploratory search: from finding to understanding. Commun ACM 49(4):41–46
71. Marenco L, Tosches N, Crasto C, Shepherd G, Miller P, Nadkarni P (2003) Achieving evolvable web-database bioscience applications using the EAV/CR framework: recent advances. J Am Med Inform Assoc 10(5):444–453
72. Margulies M, Egholm M, Altman WE, Attiya S, Bader JS, Bemben LA, Berka J, Braverman MS, Chen YJ, Chen Z et al (2005) Genome sequencing in microfabricated high-density picolitre reactors. Nature 437(7057):376–380
73. Maxam A, Gilbert W (1977) A new method for sequencing DNA. Proc Natl Acad Sci 74(2):560–564
74. Mehlhorn H, Lange M, Scholz U, Schreiber F (2012) IDPredictor: predict database links in biomedical database. J Integr Bioinform 9(2):e190
75. Murray-Rust P, Rzepa H (1999) Chemical markup, XML, and the World Wide Web. 1. Basic principles. J Chem Inf Comput Sci 39(6):928–946. http://www.xml-cml.org
76. Nolin MA, Ansell P, Belleau F, Idehen K, Rigault P, Tourigny N, Roe P, Hogan JM, Dumontier M (2008) Bio2RDF network of linked data. In: Semantic web challenge; international semantic web conference (ISWC 2008), Karlsruhe
77. O'Connor B, Day A, Cain S, Arnaiz O, Sperling L, Stein L (2008) Gmodweb: a web framework for the generic model organism database. Genome Biol 9(6):R102
78. Olivier B, Snoep J (2004) Web-based kinetic modelling using JWS online. Bioinformatics 20(13):2143–2144
79. Pearson W, Lipman D (1988) Improved tools for biological sequence comparison. Proc Natl Acad Sci USA 85:2444–2448
80. Prud'hommeaux E, Seaborne A (2008) SPARQL query language for RDF. http://www.w3.org/TR/rdf-sparql-query/
81. Richardson M, Prakash A, Brill E (2006) Beyond pagerank: machine learning for static ranking. In: WWW'06: proceedings of the 15th international conference on World Wide Web, Edinburgh. ACM, New York, pp 707–715

82. Roos DS (2001) Bioinformatics-trying to swim in a sea of data. Science 291(5507): 1260–1261
83. Saake G, Heuer A (1999) Datenbanken: Implementierungstechniken, 1st edn. MITP, Bonn
84. Sanger F, Nicklen S, Coulson AR (1977) DNA sequencing with chain-terminating inhibitors. Proc Natl Acad Sci 74(12):5463–5467
85. Schadt E, Linderman M, Sorenson J, Lee L, Nolan G (2010) Computational solutions to large-scale data management and analysis. Nat Rev Genet 11(9):647–657
86. 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
87. Schmitt I (1998) Schemaintegration für den Entwurf Föderierter Datenbanken. infix, Sankt Augustin
88. Schöch V (2001) Die Suchmaschine Google. Seminararbeit, Institut für Informatik, Freie Universität zu Berlin
89. Schönsleben P (2001) Integrales Informationsmanagement: Informationssysteme für Geschäftsprozesse – Management, Modellierung, Lebenszyklus und Technologie, 2nd edn. Springer, Berlin/Heidelberg
90. Schuler GD, Epstein JA, Ohkawa H, Kans JA (1996) Entrez: molecular biology database and retrieval system. In: Doolittle RF (ed) Computer methods for macromolecular sequence analysis. Methods in enzymology, vol 266. Academic, San Diego, pp 141–162
91. Schulz M, Krause F, Le Novère N, Klipp E, Liebermeister W (2011) Retrieval, alignment, and clustering of computational models based on semantic annotations. Mol Syst Biol 7(1):512
92. Shah S, Huang Y, Xu T, Yuen M, Ling J, Ouellette BFF (2005) Atlas – a data warehouse for integrative bioinformatics. BMC Bioinform 6(1):34
93. Siegel J (1996) CORBA fundamentals and programming. Wiley, New York
94. Siple MD (1998) The complete guide to Java database programming with JDBC. McGraw-Hill, New York/London
95. Smedley D, Haider S, Ballester B, Holland R, London D, Thorisson G, Kasprzyk A (2009) BioMart – biological queries made easy. BMC Genomics 10(1):22
96. Smith B, Ashburner M, Rosse C, Bard J, Bug W, Ceusters W, Goldberg L, Eilbeck K, Ireland A, Mungall C et al (2007) The OBO foundry: coordinated evolution of ontologies to support biomedical data integration. Nat Biotechnol 25(11):1251–1255
97. Stein L (2010) The case for cloud computing in genome informatics. Genome Biol 11(5):207
98. Stephens SM, Chen JY, Davidson MG, Thomas S, Trute BM (2005) Oracle database 10 g: a platform for BLAST search and regular expression pattern matching in life sciences. Nucl Acids Res 33(suppl_1):D675–D679
99. Taylor C, Field D, Sansone S, Aerts J, Apweiler R, Ashburner M, Ball C, Binz P, Bogue M, Booth T et al (2008) Promoting coherent minimum reporting guidelines for biological and biomedical investigations: the MIBBI project. Nat Biotechnol 26(8):889–896
100. United States National Library of Medicine (2011) Pubmed celebrates its 10th anniversary. http://www.nlm.nih.gov/pubs/techbull/so06/so06_pm_10.html
101. Valencia A (2002) Search and retrieve: large-scale data generation is becoming increasingly important in biological research. But how good are the tools to make sense of the data? EMBO Rep 3(5):396–400
102. Waltemath D, Henkel R, Winter F, Wolkenhauer O (2013) Reproducibility of model-based results in systems biology. In: Prokop A, Csukás B (eds) Systems biology: integrative biology and simulation tools. Springer, Dordrecht
103. Weiner M, Hudson T (2002) Introduction to SNPs: discovery of markers for disease. Biotechniques 32(Supplement):S4–S13
104. Weise S, Grosse I, Klukas C, Koschützki D, Scholz U, Schreiber F, Junker B (2006) Meta-all: a system for managing metabolic pathway information. BMC Bioinform 7(1):e465
105. Whetzel PL, Parkinson H, Causton HC, Fan L, Fostel J, Fragoso G, Game L, Heiskanen M, Morrison N, Rocca-Serra P, Sansone SA, Taylor C, White J, Stoeckert CJ (2006) The MGED ontology: a resource for semantics-based description of microarray experiments. Bioinformatics 22(7):866–873

106. Whetzel P, Noy N, Shah N, Alexander P, Nyulas C, Tudorache T, Musen M (2011) BioPortal: enhanced functionality via new web services from the national center for biomedical ontology to access and use ontologies in software applications. Nucl Acids Res 39(suppl_2):W541–W545
107. Wiederhold G (1996) Intelligent integration of information – foreword. J Intell Inf Syst 6(2/3):93–98
108. Wiederhold G (1997) Mediators in the architecture of future information systems. In: Huhns MN, Singh MP (eds) Readings in agents. Morgan Kaufmann, San Francisco, pp 185–196
109. Yu T, Lloyd C, Nickerson D, Cooling M, Miller A, Garny A, Terkildsen J, Lawson J, Britten R, Hunter P et al (2011) The physiome model repository 2. Bioinformatics 27(5):743–744