Mario Giacobini
Leonardo Vanneschi
William S. Bush (Eds.)

LNCS 7246

Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics

10th European Conference, EvoBIO 2012
Málaga, Spain, April 2012
Proceedings

Springer

Lecture Notes in Computer Science 7246

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Mario Giacobini Leonardo Vanneschi
William S. Bush (Eds.)

Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics

10th European Conference, EvoBIO 2012
Málaga, Spain, April 11-13, 2012
Proceedings

Springer

Volume Editors

Mario Giacobini
University of Torino
Department of Animal Production Epidemiology and Ecology
Via Leonardo da Vinci 44, 10095 Grugliasco (TO), Italy
E-mail: mario.giacobini@unito.it

Leonardo Vanneschi
Universidade Nova de Lisboa, ISEGI
1070-312 Lisboa, Portugal
and
University of Milano-Bicocca, D.I.S.Co.
Viale Sarca 336, 20126 Milan, Italy
E-mail: lvanneschi@isegi.unl.pt

William S. Bush
Vanderbilt University, Center for Human Genetics Research
Department of Biomedical Informatics
519 Light Hall, Nashville, TN 37232, USA
E-mail: william.s.bush@vanderbilt.edu

Cover illustration:
"Chair No. 17" by The Painting Fool (www.thepaintingfool.com)

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-29065-7 e-ISBN 978-3-642-29066-4
DOI 10.1007/978-3-642-29066-4
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2012933967

CR Subject Classification (1998): J.3, H.2.8, E.1, I.2, F.1, F.2.1

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

Computational biology is a wide and varied discipline, incorporating aspects of statistical analysis, data structure and algorithm design, machine learning, and mathematical modeling toward the processing and improved understanding of biological data. Experimentalists now routinely generate new information on such a massive scale that the techniques of computer science are needed to establish any meaningful result. As a consequence, biologists now face the challenges of algorithmic complexity and tractability, and combinatorial explosion when conducting even basic analyses. The goal of the 10^{th} European Conference on Evolutionary Computation, Machine Learning, and Data Mining in Computational Biology (EvoBIO 2012) was to bring together experts across multiple fields to discuss novel methods for tackling complex biological problems, and the beauty of EvoBIO is that often these experts draw inspiration from biological systems in order to produce solutions to biological problems.

The 10^{th} EvoBIO conference was held in Málaga, Spain, during April 11–13, 2012, at the Computer Science School of the University of Málaga, Spain. EvoBIO 2012 was held jointly with the 15^{th} European Conference on Genetic Programming (EuroGP 2012), the 12^{th} European Conference on Evolutionary Computation in Combinatorial Optimization (EvoCOP 2012), the First International Conference and 10th European Event on Evolutionary and Biologically Inspired Music, Sound, Art, and Design (EvoMUSART 2012), and the European Conference on the Applications of Evolutionary Computation. Collectively, the conferences are organized under the name Evo* 2012 (www.evostar.org). EvoBIO, held annually as a workshop since 2003, became a conference in 2007 and it is now the premier European event for those interested in the interface between evolutionary computation, machine learning, and data mining in computational biology. All papers in these proceedings were presented at EvoBIO 2012 in oral or poster presentations, and were received in response to a call for papers soliciting a wide range of topics in the realm of biological data analysis and computational biology.

EvoBIO 2012 added a new submission format: four-page abstracts reviewing, discussing, or extending work previously published in a journal. This form was requested in addition to our existing submission formats: 12-page full research articles describing new developments in methodology, approach, and/or application, eight-page system demonstrations of active or developing systems in research or practice domains, and eight-page short reports describing new methodologies, approaches, and/or applications. After peer review, EvoBIO accepted 15 papers for oral presentation and 8 for poster presentation.

With the goal of sharing inspiration in mind, EvoBIO and EuroGP held a special joint session to celebrate the tenth anniversary of EvoBIO. In this session, advances in the field of genetic programming were applied to problems of

computational biology, and likewise, the unique mechanisms present in biological systems were used to create new extensions to the paradigm of genetic programming. This invigorating session created new collaborations and encouraged the development of new approaches and their application to the biological problem domain.

First and foremost, we thank all the authors who spent time and effort to generate the fantastic contributions to this body of work. We thank the members of the Program Committee for their expert evaluation and review of the submitted papers. We also thank many members of the Evo* community who worked tirelessly to ensure a smooth and successful conference event; Jennifer Willies from Edinburgh Napier University for her unwavering dedication as event coordinator and the Institute for Informatics and Digital Innovation at Edinburgh Napier University; and Penousal Machado from the University of Coimbra for his fantastic work as Publicity Chair. We owe special thanks to Carlos Cotta from the University of Málaga for his outstanding planning as local organizer and for bringing Evo* to the beautiful city of Málaga. We extend our gratitude to the School of Computer Science directed by Jose M. Troya and the School of Telecommunications directed by Antonio Puerta from the University of Málaga for hosting our event, and to the Málaga Convention Bureau for their support of this conference. We also deeply appreciate the work of Marc Schoenauer from INRIA in France and the MyReview team for providing the publication management system and technical support.

We hope you enjoy the fascinating research articles included in this volume, and we invite you to contribute your work to EvoBIO 2013.

April 2012

Mario Giacobini
Leonardo Vanneschi
William S. Bush

Organization

EvoBIO 2012, together with EuroGP 2012, EvoCOP 202, EvoAPPLICATIONS 2012, and EvoMUSART 2012 was part of EVO* 2012, Europe's premier co-located events in the field of evolutionary computing.

Program Chairs

Mario Giacobini	University of Torino, Italy
Leonardo Vanneschi	Universidade Nova de Lisboa, Portugal University of Milano-Bicocca, Milan, Italy
William S. Bush	Vanderbilt University in Nashville, TN, USA

Local Chair

Carlos Cotta	University of Málaga, Italy

Publicity Chair

Penousal Machado	University of Coimbra, Portugal

Proceedings Chair

Mario Giacobini	University of Torino, Italy

Steering Committee

Elena Marchiori	Radboud University, Nijmegen, The Netherlands
Jason H. Moore	Dartmouth Medical School in Lebanon, NH, USA
Clara Pizzuti	ICAR-CNR, Italy
Marylyn Ritchie	Vanderbilt University, USA

Program Committee

Jesus S. Aguilar-Ruiz	Universidad Pablo de Olavide, Spain
Wolfgang Banzhaf	Memorial University of Newfoundland, Canada
Jacek Blazewicz	Poznan University of Technology, Poland

Erik Boczko	Vanderbilt University, USA
Ernesto Costa	University of Coimbra, Portugal
Federico Divina	Pablo de Olavide University Seville, Spain
Jitesh Dundas	Edencore Technologies, USA
Alex Freitas	University of Kent, UK
Raffaele Giancarlo	University of Palermo, Italy
Raul Giraldez Rojo	Pablo de Olavide University, Spain
Rosalba Giugno	University of Catania, Italy
Jin-Kao Hao	University of Angers, France
Tom Heskes	Radboud University Nijmegen, The Netherlands
Ting Hu	Dartmouth College, Hanover, USA
Zhenyu Jia	University of California, Irvine, USA
Mehmet Koyuturk	Case Western Reserve University, USA
Michael Lones	University of York, UK
Penousal Machado	University of Coimbra, Portugal
Bob MacCallum	Imperial College London, UK
Elena Marchiori	Radboud University, Nijmegen, The Netherlands
Andrew Martin	University College London, UK
Brett McKinney	University of Tulsa, USA
Jason H. Moore	Dartmouth College, Hanover, USA
Pablo Moscato	The University of Newcastle, Australia
Alison Motsinger-Reif	North Carolina State University, USA
Vincent Moulton	University of East Anglia, UK
Carlotta Orsenigo	Politecnico di Milano, Italy
Clara Pizzuti	ICAR-CNR, Italy
Paolo Provero	University of Torino, Italy
Michael Raymer	Wright State University, USA
Marylyn Ritchie	The Pennsylvania State University, USA
Simona Rombo	ICAR-CNR, Italy
Marc Schoenauer	LRI- Université Paris-Sud, France
Ugur Sezerman	Sabanci University, Turkey
Sara Silva	INESC-ID Lisboa, Portugal
Marc L. Smith	Vassar College, USA
El-Ghazali Talbi	Université des Sciences et Technologies de Lille, France
Marco Tomassini	University of Lausanne, Switzerland
Stephen Turner	University of Virginia, USA
Alfonso Urso	ICAR-CNR, Italy
Antoine van Kampen	Universiteit van Amsterdam, The Netherlands
Andreas Zell	University of Tübingen, Germany
Zhongming Zhao	Vanderbilt University, USA

Sponsoring Institutions

- School of Computer Science, University of Málaga, Spain
- School of Telecommunications, University of Málaga, Spain
- Málaga Convention Bureau, Spain
- The Institute for Informatics and Digital Innovationg, Edinburgh Napier University, UK

Table of Contents

Oral Contributions

Poster Contributions

Abstract Contributions

Automatic Task Decomposition for the NeuroEvolution of Augmenting Topologies (NEAT) Algorithm

Timmy Manning and Paul Walsh

Cork Institute of Technology,
Bishopstown, Cork, Ireland
timothy.manning@mycit.ie

Abstract. Neuroevolution, the process of creating artificial neural networks through simulated evolution, can become impractical for arbitrarily complex problems requiring large or intricate neural network architectures. The modular feed forward neural network (MFFN) architecture decomposes a problem among a number of independent task specific neural networks, and is suggested here as a means of managing neuroevolution for complex problems. We present an algorithm for evolving MFFN architectures based on the NeuroEvolution of Augmenting Topologies (NEAT) algorithm. The algorithm proposed here, denoted MFF-NEAT, outlines an approach to automatically evolving, attributing fitness values and combining the task specific networks in a principled manner.

Keywords: Neuroevolution, NEAT, Task Decomposition, Neural Network, Negative Correlation, Mixture of Experts.

1 Introduction

The number of neurons and synapses required by a network scales with the complexity of a problem domain. The typical approach to identifying optimal neural network architectures requires time consuming evaluations of potential architectures. Each architecture must in turn be evaluated several times starting with different initial parameter and weight configurations as typical gradient descent training algorithms can settle on local minima with suboptimal performance. For complex problems, the evaluation requirements of neural network architectures can form a bottleneck in the development process. Such problems promote the use of constructive neuroevolution algorithms for the discovery of optimal neural network architectures in an automated and methodical fashion. However, the efficiency and tractability of constructive neuroevolution approaches can also suffer as the dimensionality of the problem increases [11,8].

The NeuroEvolution of Augmenting Topologies algorithm (NEAT) and the Modular Feed Forward neural network (MFFN) architecture have desirable and complimentary features for solving complex problems. Currently, there is no

M. Giacobini, L. Vanneschi, and W.S. Bush (Eds.): EvoBIO 2012, LNCS 7246, pp. 1–12, 2012.

approach to automatically evolving optimal MFFN architectures using NEAT which maintains the advantages of both. In this paper we outline and evaluate MFF-NEAT, an approach to neuroevolving an MFFN architecture based on NEAT which:

1. Reduces a problem to a number of simpler sub-tasks
2. Promotes the dissemination of the functionality
3. Harnesses unexploited information generated in fitness evaluations

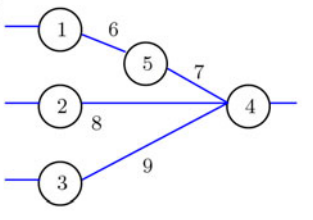
2 NeuroEvolution of Augmenting Topologies

NEAT is a leading neuroevolution algorithm which simultaneously evolves the parameters and architecture of a neural network [14]. The NEAT algorithm evolves blueprints for networks in the form of genomes which can be translated into neural networks (phenotype). The genome used in NEAT is variable length and comprises a set of neuron genes and synapse genes. The format of a NEAT gene is shown in Fig. 1(a). Fig. 1(b) shows a simple neural network and Fig. 1(c) gives the corresponding NEAT genome.

(a) Gene Format

Innovation#
Gene Type
Description
Active

(b)



(c) NEAT Genome

1	2	3	4	5	6	7	8	9
Neuron	Neuron	Neuron	Neuron	Neuron	Synapse	Synapse	Synapse	Synapse
Input	Input	Input	Output	Hidden	1 -> 5	5 -> 4	2 -> 4	3 -> 4
					True	True	True	True

Fig. 1. (a) The format of a NEAT gene. Weight values are recorded with each synapse gene, but are omitted to simplify the diagrams. (b) A sample neural network architecture. (c) The NEAT genome corresponding to the phenotype shown in (b).

The NEAT approach to evolution combines three core concepts: complexification, speciation and principled crossover. *Complexification* ensures the evolution of minimal neural networks. The initial potential network solutions contain no hidden layer neurons, with synapses linking only the input layer neurons to the output layer neurons. The hidden layer neurons and additional synapses are added over time. Structural additions to the solution which do not increase the overall fitness will tend to have little impact on future generations. In this way

the networks produced will be efficient, with minimal impact from unnecessary neurons or synapses [13].

Principled crossover is used to combat the *competing convention* problem (also known as the *permutation problem*). The competing convention problem states that between two networks learning the same problem, even of the exact same architecture, the corresponding neurons and synapses may be learning different aspects of the problem space. Therefore, crossing over genes between different genomes can lose the context of the data. Principled crossover is implemented by tracking the lineage of each gene using "*innovation numbers*". Each innovation number corresponds to a specific evolution to a network. The addition of a novel gene to a genome results in a new innovation number representing that evolution. If a gene is added to a genome which recreates a previous evolution, the previous innovation number is reused for the gene. The innovation numbers of synapses are dictated by the innovation numbers of the neurons they link. Hidden layer neurons are added splitting existing synapses, so the innovation number of a neuron is decided by the innovation number of the synapse it splits. Allowing crossover only between genes which have the same global innovation number ensures the consistent context of crossed over elements, thereby reducing the impact of the permutation problem.

Speciation is used to preserve "*innovative*" ideas created through evolution. Innovation is defined here as a sufficient evolutionary divergence in the architecture of a solution from the other solutions in the population. Such a new divergence may ultimately provide a high fitness solution, but could be discarded before it matures sufficiently for its true potential to become evident. Speciation segregates the innovative solutions in the population and guarantees them a minimum number of generations to prove their worth.

To add a new synapse, two unconnected neurons are selected. Using the genome of Fig. 1(c) as an example, the neuron genes with the innovation numbers 2 and 5 could be selected. In this situation, a synapse joining neuron 2 to neuron 5 will receive the innovation number 10. A gene representing the new synapse is added to the genome which records the innovation number and the synapse properties. Fig. 2 shows the gene which would be added to the genome representing the new synapse and the corresponding phenotype.

To add a new neuron, a synapse is first selected. Again, using the genome of Fig. 1(c) as an example, the synapse with innovation number 9 joining neurons 3 and 4 could be selected. Synapse gene 9 is marked as inactive meaning it will not be expressed in the phenotype. The new neuron is added to the network using two new synapses, joining it to the neurons to which the original synapse was connected (neurons 3 and 4 in this case). The three new genes added to the genome and the corresponding phenotype are given in Fig. 3.

3 The Modular Feed Forward ANN Architecture

Under the "*mixture of experts*" model of Jacobs and Jordan [5], a modular feed forward artificial neural network (MFFN) decomposes a task to be learned by a

(a) Additional Gene

10
Synapse
2 -> 5
True

(b)

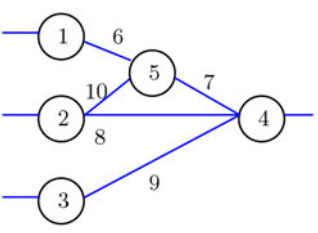


Fig. 2. (a) The gene added to the genome of Fig. 1 representing a new synapse, and (b) the corresponding phenotype

(a) Additional and Updated Genes

9	11	12	13
Synapse	Neuron	Synapse	Synapse
3 -> 4	Hidden	3 -> 11	11 -> 4
False		True	True

(b)

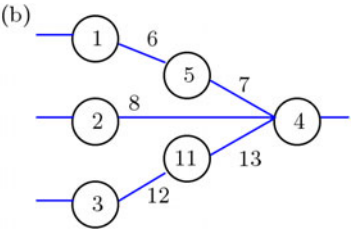


Fig. 3. (a) The updated gene (9) and 3 new genes (11, 12 and 13) representing the addition of a new neuron to the genome of Fig. 1, and (b) the corresponding phenotype

monolithic neural network into several smaller tasks. Each task is handled by a sub-network. The sub-networks are referred to as *expert networks* as each tends to focus on one particular aspect of the problem space or a reduced problem area (local computation) and works independently of the other sub-networks. In a successfully trained MFFN each expert network will compute different functions that are relevant in different scenarios or regions of the input space [5,3]. The structure of a MFFN network is given in Fig. 4.

The contribution of each expert network to the overall problem is combined by a *gating network* to form the output of the system. The gating network decides how, and under which situations to combine the outputs of the expert networks. There is however no general approach to inferring the optimal number of expert networks, the architecture of each expert network, or the optimal architecture for the gating network. The advantages of the MFFN architecture are:

1. The division of the concepts to be learned into smaller sub-networks results in an overall reduction in computational complexity of the network ("*divide-and-conquer*") [9]
2. Reduces the effect of *catastrophic interference*
3. Reduces the effect of *crosstalk*

Both catastrophic interference and crosstalk result in inefficient learning and possibly degraded performance. Catastrophic interference is caused when training data with differing output show similar patterns [17]. The MFFN architecture addresses this problem by contextually separating the processing [2,1]. Crosstalk occurs when the training of neurons suffer from trying to contribute to more than one concept, thus receiving conflicting training information. The MFFN architecture addresses this problem through the physical separation of neurons learning disparate tasks [4].

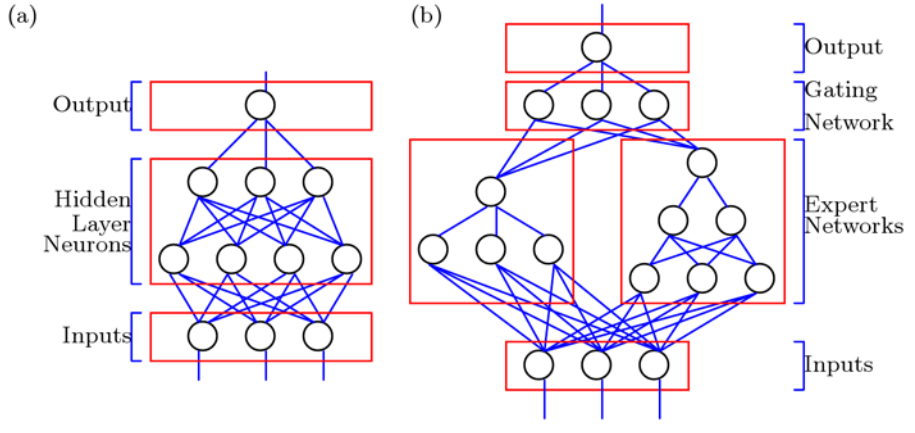


Fig. 4. (a) A fully connected monolithic MLP architecture. (b) A theoretical MFFN architecture solution for the same problem.

4 Approach

MFF-NEAT is implemented as a two-level co-evolutionary approach, with a two step sequential task decomposition [6]. MFF-NEAT combines sampling to define the functionality of expert networks and a form of disassortative (negative assortative) sexual selection to select expert networks to combine to form the MFF-NEAT systems. Disassortative sexual selection chooses parents for mating with phenotypic traits more dissimilar than likely in random mating [15]. Each MFF-NEAT system takes a subset of expert networks, defines a gating network, and produces an evaluable output. To evaluate an input vector, the vector is first applied to the phenotype of each expert. The input vector and output of the expert phenotypes are then applied to the gating network. Fig. 5 shows an example of an MFF-NEAT system.

Two separate populations are maintained; expert networks and MFF-NEAT systems. Both populations are evolved using the standard NEAT approach. Expert networks are added to the MFF-NEAT systems over time.

For each expert species and extant MFF-NEAT system a record is maintained of its fitness on each individual training exemplar. This record is referred to as

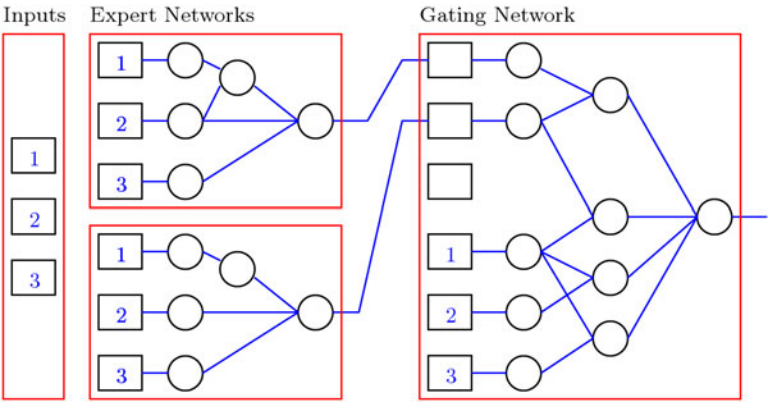


Fig. 5. A complete MFF-NEAT system. In this example, this gating network accepts as inputs the output of 2 expert networks, but can grow to accept the output of up to 3 experts.

the "*coverage vector*" and is stored as an array of fitness values. The coverage vectors are used to select complimentary expert species to add to the MFF-NEAT systems and focuses evolutionary pressure on experts which tackle specific subsets of exemplars as required. A graphical depiction of two coverage vectors and their overlap is given in Fig. 6.

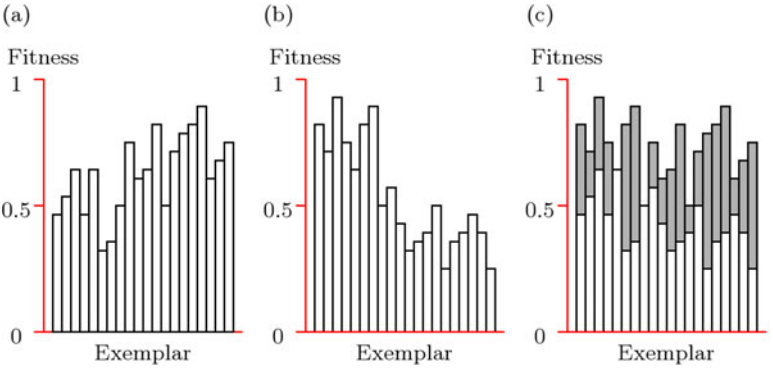


Fig. 6. Figures (a) and (b) give the theoretical coverage vectors for two expert species across 20 training exemplars. The overlap of both coverage vectors is given in (c).

A "*species archive*" is maintained of all the different expert species identified. For each expert species the archive retains a single prototype genome of the species and a coverage vector representing the average performance of experts of

that species on the training exemplars. The coverage vectors for each expert are back propagated from the evaluations of the MFF-NEAT systems which employ the experts. The coverage vectors are generated as the average of only the most elite member of each MFF-NEAT system species per generation to prevent the more populace species dominating the coverage definition.

Using the species archive and coverage vectors, extinct species can be continually re-evaluated for relevance and reintroduced at later evolutionary stages when or if they become useful, or perhaps selected to form the basis of new functionality in an exaptation-like manner. As the coverage vectors are taken by sampling from a range of constantly evolving systems comprising various combinations of the expert species, it is appreciated that this approach will not provide the archive with an exact map of which training exemplars benefit from the presence of which expert networks. Over many generations the statistical sampling approach should however result in above average peaks in the coverage vectors. These peaks should indicate the exemplars whose accurate classifications appear to benefit from the presence of the specific experts.

A novel "*disassortative evolutionary operator*" is introduced in addition to the standard NEAT operators to add new expert networks to existing MFF-NEAT systems. Expert species are selected for addition to an MFF-NEAT system based on the coverage vectors of the system itself and the coverage vectors for each expert species in the archive. The target species is the species which offers the largest theoretical increase to the coverage of the MFF-NEAT system, referred to as the "*gain*", when the coverage vectors of the MFF-NEAT system and the expert species are overlapped similar to as shown in Fig. 6(c). Once the target expert species is identified, a new expert network of this species is spawned from the expert population or the archival species prototype.

To integrate a new expert to a gating network, a new input neuron is added to the gating network. The output of this new neuron corresponds to the output of the expert network. A single synapse is added joining this neuron to the output neuron of the gating network. This approach offers minimal disruption to the learned concepts in the gating network, but the gating network must learn to successfully integrate the output of the new expert network.

5 Implementation

A population size of 150 was used for both the MFF-NEAT system and expert networks. In each generation, a new population of MFF-NEAT systems were generated from the fittest of the previous generation. Each new MFF-NEAT system has a 0.5% chance of disassortative evolution. The remaining population receive a standard NEAT evolutionary operation, with a 90% chance of a synaptic weight perturbation, a 6% chance of receiving a new synapse and a 4% chance of a new hidden layer neuron. Each system has a 50% chance of evolving it's gating network, and a 50% chance of evolving a single expert network.

An artificial limit of 4 was placed on the number of expert networks per MFF-NEAT system to prevent the systems growing large quickly and overtraining.

The complexification process of NEAT does not provide a mechanism for the removal of architectural elements and the experts of an MFFN cannot simply be switched due to the permutation problem. This results in MFF-NEAT systems which become stuck with early expert species and reach the maximum number of experts allowed quickly. To address this situation, a new MFF-NEAT system is added every 50 generations with a minimal gating network. This process allows novel combinations of expert networks in later generations and encourages small gating networks which use refined experts well.

6 Experiments

The three experiments carried out here evaluate the performance of the MFF-NEAT approach relative to standard NEAT on differing problem domains. All experiments use the same parameter sets. All experiments have a binary output, with absolute error defined as the average difference between the evidential response of the network and the expected output across all training exemplars. The MFF-NEAT experiments were run a single time, while the NEAT experiments were run five times with different initial network populations. The best performing NEAT experiment was then selected for evaluation.

6.1 Monks Third Problems

Monks data set is an artificial problem designed for the comparative evaluation of machine learning approaches [16]. Monks data set comprises six categorical attributes with 3, 3, 2, 3, 4 and 2 possible values respectively. Of the three tasks defined for this data set, only the third problem, M_3, is evaluated here as it is considered the most challenging. M_3 produces a 1 output when the fifth attribute has value 3 and the fourth attribute has value 1, or the fifth attribute does not have the value 4 and the second attribute does not have the value 3. All other conditions produce a 0 output. The data was divided into 432 training exemplars and 122 exemplars for testing. Five percent of the training data is misclassified to represent noise in the data set. The experiments were run for 10000 generations.

The MFF-NEAT approach achieved a minimum absolute error (AE) of 0.035788 at generation 1802 on the testing data. NEAT achieved 0.033628 AE at generation 4379. The peak number of correctly classified records (CCR) on the test set was 97.9167% at generation 1802 and 97.2222% at generation 3578 for MFF-NEAT and NEAT respectively.

6.2 Heart Disease Diagnosis

The goal of this data set is the diagnosis of heart disease given a set of examination results. The data used was taken from the Proben1 "*set of benchmarks and benchmarking rules for neural network training algorithms*" [10]. This data set comprises 920 exemplars with 32 input attributes. 44.7% of the exemplars

represent heart disease free cases. The exemplars were randomly divided into 670 training cases and 250 testing cases. The output values are 0 and 1, representing the absence and presence of heart disease respectively. The experiments were run for 10000 generations.

MFF-NEAT peaked at an absolute error of 0.149381 on the test set at generation 4687. The best performing standard NEAT experiment peaked at 0.1512 absolute error at generation 5725. Both approaches achieve 86% correctly classified records on the test data set.

6.3 Mass Spectral Peptide Data

The goal of this data set is to distinguish *b*- and *y-ion* series peaks in the tandem mass spectra of peptides based on the intensity of fragment ions surrounding the peak. For each *b*- and *y-ion* series peak 20 attributes were generated which describe the mass spectrum and the relative intensities of peaks at the pre-defined offsets. The data generated comprised equal numbers of randomly selected *b*- and *y-ion* series exemplars. The *b-ion* series peak are denoted with a 0 output and *y-ion* series peaks with a 1.

The exemplars were divided into three set; training data (750 exemplars), testing data (150 exemplars) and validation data (150 exemplars). For the MFF-NEAT evaluations, two thirds of the training data was allocated to generating fitness values and the remainder for generating the coverage vectors. In each generation, all population members were evaluated on the testing data set. The best performing on the testing data set was then evaluated on the validation data set to gauge the ability of the network to generalize. Six NEAT and MFF-NEAT experiments were run for 6000 generations each.

The NEAT experiments produced accuracies of 0.72236, 0.738644, 0.729544, 0.722234, 0.734529 and 0.734948 (mean 0.730). The results of the MFF-NEAT experiments were 0.733038, 0.7353, 0.738606, 0.726395, 0.751967 and 0.731926 (mean 0.736). These results were evaluated using a *Student's t-test*, to produce a t-value of 1.28. The performance of this small scale evaluation is therefore insufficient to say with confidence that MFF-NEAT offers an advantage over standard NEAT, but the results are still encouraging.

7 Discussion

MFF-NEAT produces a slightly higher absolute error than standard NEAT on the M_3 problem. It is suggested here that this failure of MFF-NEAT is attributable to the M_3 problem being insufficiently complex to benefit from task decomposition, in which case evolutionary pressure would likely offer increasing returns if spent producing a single monolithic network. The more challenging heart disease diagnosis and mass spectral data sets appear to benefit from the application of the MFF-NEAT approach in terms of absolute error. This, combined lower levels of overtraining observed in the experiments suggest that the increasingly distributed nature of the MFF-NEAT systems provide a form more

amenable to generalization. The more efficient evolution noted in the experimental data is attributed to three principle advantages identified:

1. Promotion of the dissemination of useful functionality
2. Maximization of useful information generated in evaluations
3. Reduction in the complexity of the functions evolved

7.1 Promotion of the Dissemination of Useful Functionality

The independent nature of the expert networks makes them a favourable approach to neuroevolution. This independence facilitates the propagation of the expert functionality among MFF-NEAT systems of varying species in a highly principled crossover-like manner without disruption to previously learned concepts and avoiding the competing conventions problem. The disassortative evolutionary operator and the use of coverage vectors influence the evolution of the expert population in a way which produces complimentary networks which work well together. Additionally, networks are evolved in parallel under different conditions (in different combinations of experts) keeping diversity high in the population and increasing the chance of discovering useful expert networks.

7.2 Maximization of Useful Information Generated in Evaluations

Standard NEAT evaluates the potential solution populations on every exemplar, but uses only the average fitness in selecting parents from which to generate the following generations. The coverage vectors maintained by MFF-NEAT record and use the classification ability of potential solution networks and expert species on a per-exemplar basis. The expert species coverage vectors are retained and refined over the lifetime of the program. The overlaps in the solution structures and the coverage vectors are used to attribute specific functionality to the expert networks. As the sampling size for the coverage increases, an increasingly accurate impression of the ability of the experts is achieved, resulting in further refinement to the principled nature of the disassortative evolutionary operator.

7.3 Reduction in the Complexity of Functions Evolved

The principle of *divide and conquer* reduces a problem to into a number of subproblems, each tackled by an independent network. Each network will tend to deal with only a subset of the input attributes and require only a fraction of the neurons and synapses of a monolithic neural network attempting to solve the entire problem. This results in small genomes for the expert networks and a simplified error surface [12], meaning specific (but reduced) functionality can be evolved efficiently. The isolation of the task specific neurons and synapses in the expert networks has the additional effect of minimizing the potential for crosstalk and catastrophic interference, further reducing the complexity of the neuroevolution process. Similar benefits can be expected for the MFFN gating network, as it does not need to relearn functionality which has been "*farmed out*" to the expert networks.

The use of negative correlation for neuroevolution is not a novel idea. A similar approach in this area is that of Liu *et al.* [7], referred to as *EENCL*, which employs a negative correlation to evolve different weight sets for the training exemplars. Networks trained using different weights have disparate evolutionary focuses and tend to work well together when joined as an ensemble. Although similar, the MFF-NEAT approach offers a number of advantages over EENCL. The coverage vectors allow MFF-NEAT to identify patterns and functionality which are useful only in an indirect manner to the output of the network, which facilitates the two-step sequential task decomposition possible through the use of the MFFN architecture. The coverage vectors also provide a means of selecting apt sub-networks to combine. The gating network of the MFFN architecture also offers a method for intelligently combining the outputs of the sub-networks. It is also considered that the favourable ensemble properties of the EENCL approach could be emulated in MFF-NEAT through allowing an excess of experts of the same species or of similar functionality.

8 Conclusions and Future Work

In this paper we describe and evaluate MFF-NEAT, a novel approach to neuroevolution. MFF-NEAT takes advantage of the speciation and complexification of the NEAT algorithm to provide an efficient means of evolving modular neural network solutions. MFF-NEAT can automatically decompose a task and produce expert networks which encode functional points or pattern recognitions which may be either directly or indirectly relevant to the output of the gating networks. The concepts of a species archive and coverage vectors are introduced to optimize the usage of data generated in fitness evaluations, encourage the re-use of previously defined functionality and allow the focusing of evolutionary pressures on specific insufficiencies of a solution. In evaluations, MFF-NEAT is shown to offer a general performance advantage over standard NEAT on many problems.

Future work on this project will be focused on evaluating and expanding the scalability of MFF-NEAT. Scalability will be evaluated through application to complex large scale real world bioinformatics problems. The ability of MFF-NEAT to handle noisy data will be evaluated through the use of simulated complex data sets with controllable levels of noise.

References

1. Berthouze, L., Tijsseling, A.: A neural model for context-dependent sequence learning. Neural Process. Lett. 23, 27–45 (2006)
2. French, R.M.: Grid Information Services for Distributed Resource Sharing. In: Sixteenth Annual Conference of the Cognitive Science Society, pp. 335–340. Routledge (1994)
3. Happel, B.L.M., Murre, J.M.J.: Design and evolution of modular neural network architectures. Neural Networks 7, 985–1004 (1994)

4. Jacobs, R.A., Jordan, M.I., Barto, A.G.: Task decomposition through competition in a modular connectionist architecture: The what and where vision tasks. Cognitive Sci. 15, 219–250 (1991)
5. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural Comput. 3, 79–87 (1991)
6. Khare, V.R., Yao, X., Sendhoff, B., Jin, Y., Wersing, H.: Co-evolutionary modular neural networks for automatic problem decomposition. In: IEEE Congress on Evolutionary Computation, pp. 2691–2698. IEEE (2005)
7. Liu, Y., Yao, X., Higuchi, T.: Evolutionary Ensembles with Negative Correlation Learning. IEEE Trans. Evol. Comput. (2000)
8. Mouret, J.B., Doncieux, S.: Evolving modular neural-networks through exaptation. In: IEEE Congress on Evolutionary Computation, pp. 1570–1577. IEEE (2009)
9. Panait, L.: Theoretical convergence guarantees for cooperative coevolutionary algorithms. Evol. Comput. 18, 581–615 (2010)
10. Prechelt, L.: Proben1: A set of neural network benchmark problems and benchmarking rules. Fakultät für Informatik, Univ. Karlsruhe, Karlsruhe, Germany, Tech. Rep. 21 (1994)
11. Reisinger, J., Stanley, K.O., Miikkulainen, R.: Evolving Reusable Neural Modules. In: Deb, K., et al. (eds.) GECCO 2004. LNCS, vol. 3103, pp. 69–81. Springer, Heidelberg (2004)
12. Shang, Y., Wah, B.W.: Global optimization for neural network training. Computer 29, 45–54 (1996)
13. Stanley, K.O., Miikkulainen, R.: Competitive coevolution through evolutionary complexification. Artif. Intell. Res. (JAIR) 21, 63–100 (2004)
14. Stanley, K.O., Miikkulainen, R.: Evolving neural networks through augmenting topologies. Evol. Comput. 10, 99–127 (2002)
15. Staub, J.E.: The Grid: Crossover: Concepts and Applications in Genetics, Evolution, and Breeding: An Interactive Computer-Based Laboratory Manual. University of Wisconsin Press (1994)
16. Thrun, S.B., Bala, J.W., Bloedorn, E., Bratko, I., Cestnik, B., Cheng, J., De Jong, K., Dzeroski, S., Fahlman, S.E., Fisher, D., Hamann, R., Kaufman, K., Keller, S., Kononenko, I., Kreuziger, J., Michalski, R.S., Mitchell, T.M., Pachowicz, P., Reich, Y., Vafaie, H., Van de Velde, W., Wenzel, W., Wnek, J., Zhang, J.: The MONK's problems: A Performance Comparison of Different Learning Algorithms. Computer Science Reports, CMU-CS-91-197, Carnegie Mellon University, Pittsburgh, PA (1991)
17. Wiskott, L., Rasch, M.J., Kempermann, G.: A functional hypothesis for adult hippocampal neurogenesis: avoidance of catastrophic interference in the dentate gyrus. Hippocampus 16, 329–343 (2006)

Evolutionary Reaction Systems

Luca Manzoni[1], Mauro Castelli[1], and Leonardo Vanneschi[2,1]

[1] Dipartimento di Informatica, Sistemistica e Comunicazione (DISCo)
Univesità degli Studi di Milano-Bicocca, Viale Sarca 336, 20126 Milano, Italy
{luca.manzoni,mauro.castelli,vanneschi}@disco.unimib.it
[2] ISEGI, Universidade Nova de Lisboa, 1070-312, Lisboa, Portugal
lvanneschi@isegi.unl.pt

Abstract. In the recent years many bio-inspired computational methods were defined and successfully applied to real life problems. Examples of those methods are particle swarm optimization, ant colony, evolutionary algorithms, and many others. At the same time, computational formalisms inspired by natural systems were defined and their suitability to represent different functions efficiently was studied. One of those is a formalism known as reaction systems. The aim of this work is to establish, for the first time, a relationship between evolutionary algorithms and reaction systems, by proposing an evolutionary version of reaction systems. In this paper we show that the resulting new genetic programming system has better, or at least comparable performances to a set of well known machine learning methods on a set of problems, also including real-life applications. Furthermore, we discuss the expressiveness of the solutions evolved by the presented evolutionary reaction systems.

1 Introduction

It is nowadays about fifty years since the very first computational experiments that originated Genetic Programming (GP) and about twenty years since John Koza named and popularised the method [14]. During the past two decades there has been a significant range and volume of development in the theory and application of GP and GP is nowadays recognized as a well established research field [21]. Large part of the efforts of researchers has been dedicated to the study of the evolution of several different computational formalisms, that can help practitioners to solve problems with different levels of expressiveness. Under this perspective, from the very earliest experiments in the automatic generation of executable structures [6] a variety of representations have been explored starting with binary string machine code [9], finite state automata [7], generative grammatical encodings [24] to the dominant tree-based form popularised by Koza [14]. To this day numerous alternative representations have been proposed including graph [23], strongly-typed [16], linear-tree [12], and linear-graph [13]. Among the many variants, particularly popular are the developments in grammar-based GP (see for instance [17]) and cartesian GP (see for instance [15]). Besides [21], the interested reader is referred to [18] for an in-depth discussion of the open issues opened by the several different GP representation models that have been proposed along the years.

This paper is situated in this vast research field, and its aim is the one of proposing a new GP system, able to evolve programs expressed in a new and challenging computation formalism called *Reaction Systems* (RS) and recently introduced by Rozenberg

M. Giacobini, L. Vanneschi, and W.S. Bush (Eds.): EvoBIO 2012, LNCS 7246, pp. 13–25, 2012.

and coworkers [3] (an introduction to RS is offered in Section 2). This new GP variant will be called *Evolutionary Reaction Systems* (EvoRS).

Why introducing a new GP variant, evolving another computational formalism, despite the many variants already defined so far? Many answers could be given to this question, justifying the fact that evolving RS is interesting and relevant. First of all, as it will be clear in the continuation of this paper, RS is a powerful and expressive computation formalism, that is particularly intuitive, being based on a single, simple concept: the one of *reaction* (clearly inspired by chemical reactions). As such, RS can simulate constructs that allow to produce non-terminating programs (like iterations or recursion) without explicitly using them. Another reason why the introduction of EvoRS is, in our opinion, relevant is that it lightens the final user from the burden of defining the set of functional symbols used to build up the evolved programs. This definition is clearly a crucial step in many of the most currently used GP variants, including tree based GP and grammar-based GP, since it has a direct impact on the ability of the GP system to find good solutions and it must be completely hand-defined by the final user. Last but not least, RS is a bio-inspired computational formalism, and in [18], O'Neill and coworkers dedicate an entire section of their GP open issues chapter to "The Influence of Biology on GP", claiming that we currently do not use a sufficient set of features from biological evolution to embody its full potential in our artificial evolutionary process, and that in order to provide GP with new potentials and power we need to go back to the natural example of biology and to study what else can be learned from it. This paper is intended to represent a step in this direction. Not only our objective is introducing EvoRS and discussing its functioning, but we also want to give an idea of the potentialities of this new evolutionary algorithm, by comparing its performances with the ones of other well known machine learning methods (including standard tree-based GP) and by discussing the expressiveness of its returned solutions.

This paper is structured as follows: in Section 2 we introduce RS, describing their functioning and discussing relevant bibliographic material; Section 3 presents EvoRS, illustrating the main ideas behind it, and its composing elements; Section 4 discusses the test problems that we have used to validate EvoRS and the experimental settings and obtained results; finally Section 5 concludes the paper.

2 Preliminary Notions

Reaction Systems. Reaction systems were introduced by Rozenberg and Ehrenfeucht [3] in 2004 as a formalism inspired by chemical reactions. The model was defined to be simple and easily extensible. Indeed, in the recent years it has been extended to include, for example, the notion of time [5] which was not present in the original formulation (the one we are considering here). Thus, the model can be adapted to different needs, providing both a formalism that allows to study the formal properties and a modeling tool for real-life chemical systems. In this section we provide a brief explanation of the necessary notions about reaction systems. For a complete introduction, refer to [3]. A central concept in reaction systems is the one of *reaction*. Every reaction is composed by a set of reactants that are necessary for the reaction to happen, a set of inhibitors whose aim is blocking the reaction and a set of chemicals that are produced by the reaction.

Definition 1. *A* reaction α *is a triple of non-empty sets* (R, I, P) *with* $R \cap I = \emptyset$*. The set* R *is called the set of* reactants*,* I *is the set of* inhibitors *and* P *is the set of* products*. Let* S *be such that* $R, I, P \subseteq S$*, then* α *is a reaction on* S*.*

The set of all reactions that can be defined over a set S is denoted by $\text{rac}(S)$. Given a set $T \subseteq S$ and a reaction $\alpha = (R_\alpha, I_\alpha, P_\alpha) \in \text{rac}(S)$, α is *enabled* on T iff $R_\alpha \subseteq T$ and $I_\alpha \cap T = \emptyset$ (i.e., all the reactants and none of the inhibitors are present). The result of α on T, denoted by $\text{res}_\alpha(T)$, is P_α if α is enabled on T and $\emptyset$ otherwise. Note that no set of the triple that define a reaction can be empty. This modeling was chosen to better simulate real systems: every reaction needs some reactant to transform, can be inhibited in some way and produces some chemicals. Equipped with the concept of reaction we can now define what a reaction system is. In fact, it can be considered as a set of reactions that can act over a certain set of chemicals.

Definition 2. *A* reaction system*, RS from now on,* $\mathcal{A}$ *is a pair* (S, A) *where* S *is a finite set of symbols and* $A \subseteq rac(S)$*.*

The notion of results of A on T (or, equivalently, of the entire RS $\mathcal{A}$ on T), with $T \subseteq S$, is $\bigcup_{a \in A} \text{res}_a(T)$ and it is denoted by $\text{res}_T(A)$. After introducing the concept of results, it is necessary to define the dynamics of a RS. The dynamics of a RS is given by a pair of finite sequences of sets (i.e., an iterative process) that represents, step by step, the chemicals present in the system and the ones that are added from outside.

Definition 3. *Let* $\mathcal{A} = (S, A)$*. An* iterative process π *is a pair of finite sequences* (γ, δ) *with* $\gamma = C_0, C_1, \ldots, C_n$ *and* $\delta = D_1, \ldots, D_n$ *of subsets of* S*. Furthermore,* $D_1 = res_{\mathcal{A}}(C_0)$ *and* $D_i = res_{\mathcal{A}}(D_{i-1} \cup C_{i-1})$ *for all* $1 < i \leq n$*.*

The sequence γ of an iterative process represents the chemicals inserted at every time step from outside. The sequence δ represents the chemicals that are produced by the RS. The state of a RS is given by the finite sequence of sets $W_0, \ldots, W_n$ with $W_0 = C_0$ and $W_i = C_i \cup D_i$ for all $1 \leq i \leq n$. The set W_0 is called the initial state of the system (i.e., the set of chemicals initially present in the system). The set W_i for $0 \leq i \leq n$ is called the state of the system at time i. We will consider only RS such that $C_i = \emptyset$ for all $1 \leq i \leq n$. In this case the dynamics is necessarily ultimately periodic [4] (i.e., it necessarily reaches a loop where we are cycling between a finite number of set of chemicals). There is a natural partial ordering between reactions. Let $a, b \in \text{rac}(S)$, then $a \leq b$ iff $\text{res}_T(a) \subseteq \text{res}_T(b)\ \forall T \subseteq S$. It has been proved [4] that this is equivalent to $R_a \supseteq R_b$, $I_a \supseteq I_b$ and $P_a \subseteq P_b$. This property allows us to easily simplify a RS by removing unnecessary reactions (this action will be performed by a particular genetic operator). This partial ordering can be used to find a system equivalent (i.e., that has the same behaviour with all inputs) to a given one but with less reactions. In fact, it is possible to prove that every set of reactions A is equivalent to its subset that contains only all the maximal elements of A. This means that we can easily simplify a RS by removing the reactions that do not influence its dynamics.

Example 1. Boolean functions can be easily represented by RS. As an example consider the *and* function with two inputs. It can be represented as a reaction system with $S = \{x_1, x_2, \text{True}, i\}$, where x_1 and x_2 are the input variables, True is a constant that

represents the output *true* of the system and i is a dummy inhibitor (a symbol that can only be in an inhibitor set and is never inserted in the system nor produced by other reaction). The set of reactions A contains only the reaction $\alpha = (\{x_1, x_2\}, \{i\}, \{\text{True}\})$. The outputs of the system with all possible inputs are the following:

$$\{x_1\} \longrightarrow \emptyset \qquad \{x_2\} \longrightarrow \emptyset \qquad \emptyset \longrightarrow \emptyset \qquad \{x_1, x_2\} \xrightarrow{\alpha} \{\text{True}\}$$

where on the left of the arrow there is the initial state, on the right the state at time 1 and the superscript over the arrow indicates the reactions that were enabled. This notation means that, if we have only x_1 in the system, no reaction is enabled and hence we obtain the empty set as a result. If only x_2 is present in the system then we also obtain the empty set as a result. Moreover, when we have no symbols in the system we do not generate other symbols. Finally, when both x_1 and x_2 are present in the system the reaction α is enabled (indicated by the superscript over the arrow) and we generate the symbol True. Denoting a true variable by inserting its corresponding symbol in the initial state of the system and a false variable by not inserting it, the reaction α clearly represents an *and* gate. Also notice that it is always possible to insert a dummy inhibitor (a symbol that is never present) and a dummy reactant (a symbol that is always present) in order to avoid the use of empty sets in the definition of reactions. Therefore, we will allow empty sets either as reactant sets or as inhibitor sets since they can be easily simulated using dummy symbols.

3 Evolutionary Reaction Systems

In this section an evolutionary version of reaction systems is presented. We will call it *Evolutionary Reaction Systems* (EvoRS). An EvoRS individual is a RS. A population is a set of RS. We will discuss only the phases of EvoRS that are different from others evolutionary algorithms. For example, selection, being based on the phenotype, it does not depend on the particular representation used and then could be performed using one of the standard algorithms (i.e., roulette-wheel, tournament, etc.).

Input and Output for EvoRS. One first aspect to note is that it is necessary to allow both input and output from a RS. Let $x_1, \ldots, x_m$ be the set of input variables. Suppose that every variable can assume only a finite number n_i of values: $x_i \in \{k_1, \ldots, k_{n_i}\}$. Fix $i \in \{1, \ldots, m\}$. To the variable x_i we will associate $n_i - 1$ input symbols, that represents the predicates $x_i = k_2, \ldots, x_i = k_{n_i}$. The predicate $x_i = k_1$ will be represented by the absence of an input symbol. Thus, there will be $\sum_{i=1}^{m}(n_i - 1)$ input symbols. An important characteristic of RS is that the symbols that are not explicitly preserved from one step to the other disappears (e.g., in 1 the symbol True is not preserved, hence we can have it at $t = 1$ but not at $t = 2$). In fact, using a set of output symbols, if one of them appears at time t, we are not assured that in the subsequent time steps it will be preserved. Therefore, we decided to fix a parameter `execution length` that is a positive natural number. Suppose we have $o_0, o_1, \ldots, o_n$ possible output values. Choose o_0 as a default value. We will prevent it from being generated by any reaction. We run the RS for `execution length` steps and, if during the execution one of the output symbols $o_1, \ldots, o_n$ is produced then it is returned as output. Otherwise the

default value o_0 is returned. In this way we are certain that a RS will always produce an output. Hence, a fitness evaluation can be performed.

Initialization. The initialization of an EvoRS is simply the creation of a population of randomly generated RS. Three parameters are needed: (1) the `population size`, (2) the `number of symbols` that can be used in the system (note that they need to be at least as many as the input symbols plus the output symbols); (3) the maximum `initial size` (when a RS is randomly generated, the number of reactions that it contains – i.e., its size – is chosen randomly but its value is bounded by the `initial size` parameter).

Crossover. Given the list of individuals $\mathcal{A}_1, \ldots, \mathcal{A}_n$ obtained by selection, for any pair $\mathcal{A}_{2i-1}, \mathcal{A}_{2i}$ with $1 \leq i \leq \frac{n}{2}$, there is a probability p_c that it will be subject to crossover. Given two RS, $\mathcal{A} = (S, A)$ and $\mathcal{B} = (S, B)$ a *crossover* of $\mathcal{A}$ and $\mathcal{B}$ is a stochastic operator that generates two reactions systems $\mathcal{A}'$ and $\mathcal{B}'$ in the following way: (1) let $C = A \cup B$; (2) let $k \in \{1, \ldots, |C| - 1\}$; (3) let A' be a subset of cardinality k of C (it can be randomly selected, or, if the element of C are ordered, it is possible to take the first k elements); we define: $\mathcal{A}' = (S, A')$ and $\mathcal{B}' = (S, C \setminus A')$. Note that since $A \cap B$ can be non-empty, we have that $|A| + |B| \geq |A \cup B| = |A'| + |B'|$, thus possibly reducing the total number of reactions.

Mutation. We have defined three kind of mutation for RS. One type of mutation is the *random reaction insertion*. Fix a RS $\mathcal{A} = (S, A)$. With probability p_{in} for any of the reactions in A, another randomly generated reaction is inserted. The second type of mutation is the *random reaction removal*. Let $\mathcal{A} = (S, A)$ be a RS. All reactions in A have probability p_{rm} of being removed. The last kind of mutation, called *renewal*, is, in fact, a random recreation of the RS with probability p_{ren}. The system that is generated has a number of reactions chosen randomly and bounded by the `initial size` parameter.

Minimization of Reaction Systems. To simplify the individuals that are in the population, we introduce another genetic operator, that we call *minimization*, that reduces the number of reactions that comprises a RS without altering its behaviour, by eliminating reactions that have no impact on the results (equivalent of "dead code"). This operator is always applied and is based on the following observation: given a RS $\mathcal{A} = (S, A)$, the set $B \subseteq A$ of all maximal reactions in A can replace A without altering the behaviour of the system. After the introduction of all the genetic operator we can recall all the parameters that are necessary for EvoRS (see Table 1).

There are four parameters related to the entire system (from the population size to the length of the execution of a RS). There is one parameter for crossover and three parameters for the three different types of mutation. The operators are applied as specified by Fig. 1. After the selection phase the more destructive kind of mutation, the *renewal*, is applied. The next operator to be applied is the crossover, followed by the two less destructive kind of mutation. The EvoRS cycle is ended by the application of the *minimization* operator.

Table 1. The parameters of an EvoRS system

Parameter	Meaning
`population size`	Number of RS in the population
`number of symbols`	Number of symbols used in the creation of reactions
`initial size`	Initial size (i.e., number of reactions) of the RS
`execution length`	Number of iterations to perform during the fitness evaluation
p_c	Crossover probability
p_{in}	Probability of inserting a random reaction into a system
p_{rm}	Probability of removing a random reaction from a system
p_{ren}	Probability of regenerate randomly the current RS

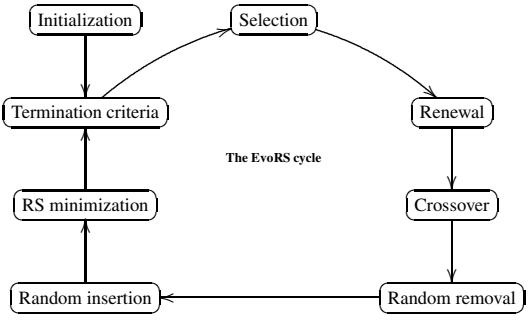


Fig. 1. The execution cycle of EvoRS

Properties of EvoRS. Before describing the experimental results of EvoRS, it is interesting to note some of the advantages that they may have compared to other machine learning techniques. First of all, RS are not black boxes: their reactions, and the interactions between them, can be read and interpreted by humans. A second advantage with respect to other techniques, as GP, is that it is not necessary to define a set of functional and terminal symbols specific to the problem under exam. In fact, only the number of symbols used by the reactions is a necessary parameter. The operations are carried on by the reactions an the interactions between them. As currently defined, EvoRS also has a disadvantage: since for any input variable it is necessary to have a number of symbols comparable to the number of values that the variable can assume, we cannot use EvoRS on problems with continuous variable. An extension of EvoRS capable of handling this kind of problems is currently under study and will be presented in the future.

4 Experimental Study

To validate EvoRS, we compared it to some well-known machine learning methods on three different test problems. In this section we present the test problem used, then we will briefly introduce the other machine learning method used. Finally, the experimental settings are described.

Test Problems. We report the problems used in the experimental phase. All the considered test problems concern the classification of instances in two target classes. The first

problem is the well known k-even parity problem (see [14] for a definition of this problem). In the experimental phase we considered binary sequences of length from 2 to 8. In the second test problem the task is to distinguish democrat votes from republican votes in the 1984 United States Congressional Voting Records. The data are the position taken by the representative on 16 key votes identifies by the Congress Quarterly Almanac. The dataset (available in WEKA [10]) has 435 instances and 17 attributes (where the last attribute is the target class). All the non-target attributes assume three possible values: *yes*, *no* and *unknown* (corresponding to a position that is neither *yes* or *no*). The target attribute assumes only two values: *republican* and *democrat*. The last test problem regards diagnosing of cardiac Single Proton Emission Computed Tomography (SPECT) images. Each of the patients is classified into two categories: *normal* and *abnormal*. The dataset has 267 instances that are described by 23 binary attributes (where the last attribute is the target class). Each row of the dataset represent an image of a different patient, where the attributes are the result of a processing that extracted 44 continuous features that describe the image, 22 of whom were selected in a subsequent phase. This dataset is available at the UCI Machine Learning Repository [8].

Other Studied Techniques. In the experimental phase, performances obtained with EvoRS have been compared with the results obtained considering different machine learning techniques. The machine learning techniques chosen to make the comparison are: feed-forward artificial neural networks trained with back-propagation (ANN), Bayesian networks (Bayes Net), naive Bayes classifier (Naive Bayes), radial basis function networks (RBF Net), and support vector machines using the sequential minimal optimization algorithm (SVM/SMO). For a complete description of these methods we refer to [1] for ANN, to [11] for Bayes Net, to [22] for Naive Bayes, to [19] for RBF net and to [2] and [20] for SVM/SMO. Furthermore, we compared EvoRS standard tree-based Genetic Programming (GP) [21].

Experimental Setting. Due to the fact that both deterministic and non-deterministic machine learning methods are used in the experimental phase, it is important to explain how the experiments have been performed in order to produce a fair comparison of the results. Fitness was calculated as the number of correctly classified instances. We used the same set for both training and testing the algorithm. Hence, in these tests, we are not concerned with the issues of overfitting or generalization ability.

For all the non-evolutionary techniques we used the implementation of WEKA [10]. For the two Bayesan techniques (Bayes Net and Naive Bayes) only one run using the default WEKA's parameter setting has been performed. For ANN, RBF Net and SVM/SMO, 100 independent runs using the default WEKA's parameters have been performed. In each run we changed the seed used to generate random numbers in the algorithm. For GP, 100 independent runs have been performed for each of the considered test problems. All the runs used populations of 100 individuals allowed to evolve for 100 generations. Tree initialization was performed with the Ramped Half-and-Half method [21]. The maximum initial depth was 4 for the SPECT and voting datasets, while it was equal to k for k-even parity problems. The function set contained the three boolean operators *and*, *or*, and *not*. For the vote dataset they where interpreted as three-valued logic operators (i.e., the conjunction of a true or unknown value with an unknown value gives an unknown value, the negation of an unknown value remains unknown and

the disjunction of a false or unknown value with an unknown value remains unknown). When the evaluation of a tree returned *unknown* instead of a specific class, its value was considered equal the one of the most represented class. The terminal set contained a number of variables equal to the number of attributes of each test problem. We have explicitly imposed functions and terminals to have the same probability of being chosen when a random node is needed. The reproduction (replication) rate was 0.1. Standard tree mutation and standard crossover (with uniform selection of crossover and mutation points) were used with probabilities of 0.1 and 0.9, respectively. The new random branch created for mutation has maximum depth 4. Selection for survival was elitist, with the best individual preserved in the next generation. The maximum tree depth is 17 except for the even parity problem, where the maximum tree depth is $2k$.

For EvoRS, 100 independent runs allowed to evolve for 100 generations were performed. In all the run elitism was used. Hence, the individual with the best fitness was preserved across generations. For all the problems a population size of 50 individuals was used (half of the population size with respect to GP). The number of symbols was two times the number of input variables for the k-even parity problem and the SPECT problem. For the *vote* dataset we used two times the number of input variables plus 10 additional symbols. This variation was necessary since every input variable in the *vote* dataset can assume three values instead of two. The initial size of the RS was two times the number of input variables. For all the problems the execution length was 3. A crossover probability of 0.8 was chosen. The probability of a random insertion was fixed to 0.2. The probability of random removal was fixed to 0.2 and the probability of renewal was 0.1 for all the considered problems. Furthermore, elitism was used. It is important to note that a research of the best set of parameters has not been performed. Hence, these parameters need to be considered a guess based on a very limited number of test runs. A more detailed explanation of the parameters setting will be the focus of successive researches.

Experimental Results. In this section the results of the experimental phase are presented. Furthermore, an example of the structure of the solutions generated by EvoRS is presented. All the box plots presented have the end of the two whiskers representing one standard deviation above and below the mean of the data. The cross represents the mean of the data. The fraction of successfully classified instances for GP and EvoRS is the one obtained after the last considered generation.

k-Even Parity. The results for the k-even parity problem are presented in Fig. 2. EvoRS and GP perform better than the other considered techniques for all the tested values of k. In particular, EvoRS is the best performer for values of k between 2 and 5, while GP performs better for values of k greater than 5. To test whether or not the differences in terms of fitness between the considered techniques are statistically significant, a test of statistical significance has been performed. First of all, a Kolmogorov-Smirnov (KS) test with a significance level of $\alpha = 0.05$ has been performed to test whether or not the fitness values are normally distributed. The (KS) test rejects the null hypothesis (hence the fitness values are not normally distributed) for all the k values and for all the non-deterministic techniques. Because the data are not normally distributed, a rank-based statistic has been used. The Wilcoxon rank-sum test for pairwise data comparison with a Bonferroni correction for the value of α is used under the alternative hypothesis

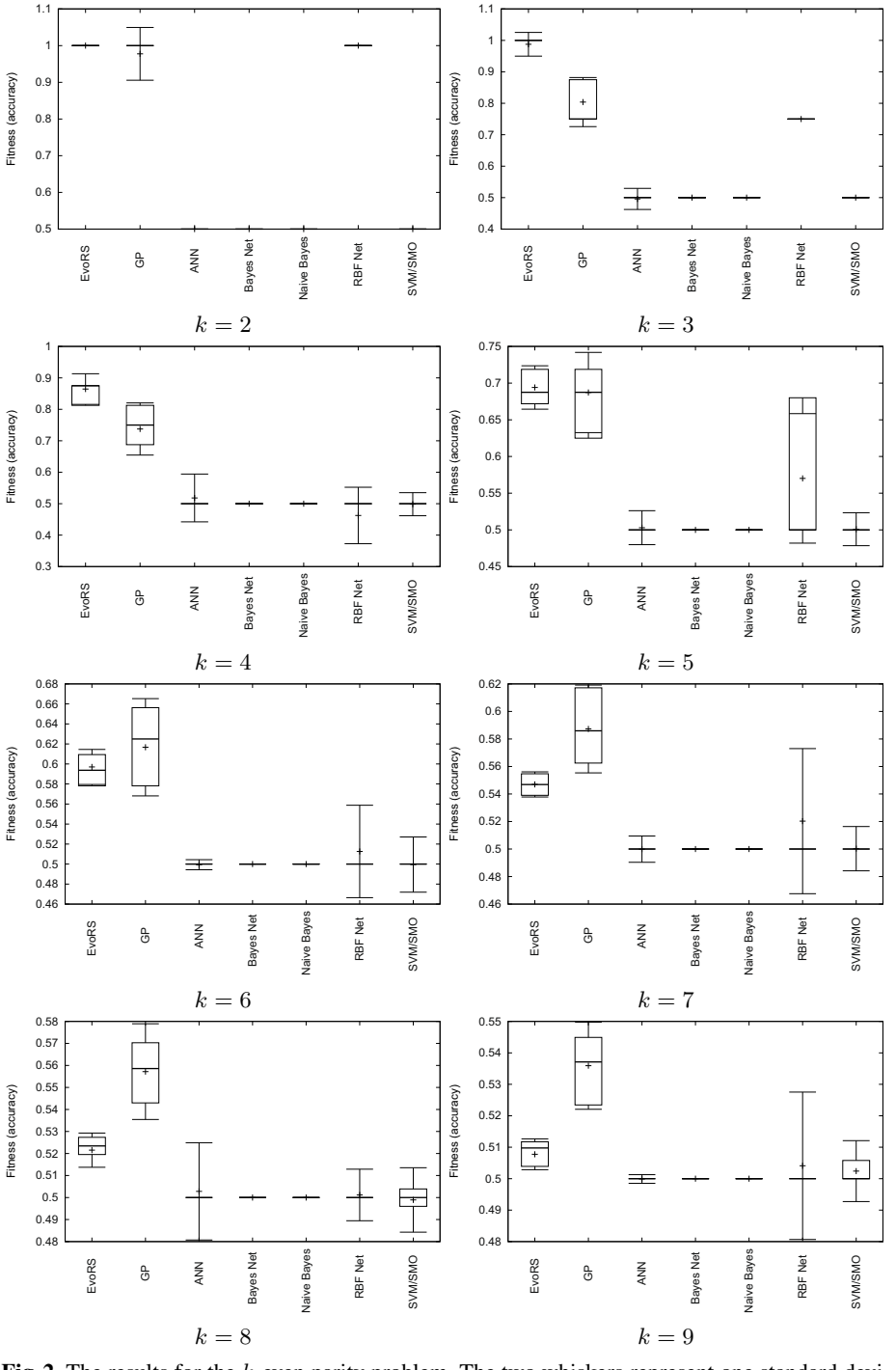
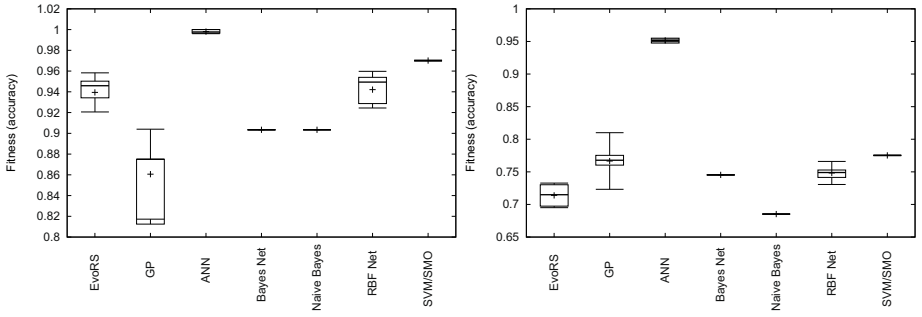


Fig. 2. The results for the k-even parity problem. The two whiskers represent one standard deviation above and below the mean of the data.

that the samples do not have equal medians. The test has been performed by comparing EvoRS with the other techniques. We obtained that we can not reject the null hypothesis only in three cases: with $k = 2$ when comparing EvoRS with GP and RBF Net and also when $k = 5$ when comparing EvoRS with GP. In all the other cases the presented results have a statistically significant difference.

Vote Dataset. The results for the vote problem are presented in Fig. 3(a). In this case, ANNs is the best performer with the 99% of correctly classified instances. SVM/SMO produce a 97% of correctly classified instances followed by EvoRS and RBF Net with 94%. It is important to underline that EvoRS is a newly defined evolutionary technique, hence the tuning of its parameters is quite difficult. Nonetheless it produces results that are comparable with the ones produced by other well-known (and well studied) machine learning techniques. Also for this problem the same statistical tests as for the k-even parity have been performed. The null hypothesis cannot be rejected only when comparing EvoRS with RBF Net.



Vote Dataset (a) Spect Dataset (b)

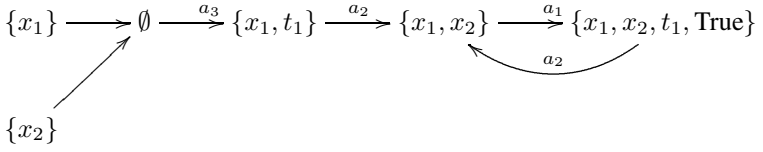
Fig. 3. The results for the *vote* and SPECT datasets. The two whiskers represent one standard deviation above and below the mean of the data.

SPECT Dataset. Results for the SPECT dataset are presented in Fig. 3(b). In the SPECT problem EvoRS performance is lower than the other contenders except Naive Bayes. The same statistical tests performed for the other test problems have been considered for this problem. From the results of the Wilcoxon test we obtained that, for this problem, the null hypothesis has been always rejected. So difference in performance of the different studied methods is statistically significant. Nonetheless EvoRS performances are non very low compared to the other techniques. We do not consider this a negative result for a newly-developed and still-to-be-tuned evolutionary algorithm.

Some Individuals Found by EvoRS. Contrarily to many other machine learning techniques and similarly to GP, EvoRS provides models that are directly interpretable by humans. Let us consider, for instance, a solution generated by EvoRS for the even parity problem with $k = 2$. It is composed by the following three reactions:

$$a_1 = (\{x_1, x_2\}, \{t_1\}, \{x_1, x_2, t_1, \text{True}\})$$
$$a_2 = (\{t_1\}, \emptyset, \{x_1, x_2\}) \qquad a_3 = (\emptyset, \{x_1, x_2, t_1\}, \{x_1, t_1\})$$

Where x_1 and x_2 are the input symbols, True is the output symbols representing *true* and t_1 is a temporary symbol. The dynamic evolution of the system can be represented by the following graph, where the nodes are states and the transitions between them are represented by the edges labeled with the reactions that are activated:



Recall that the test were performed with an `execution length` of three. Hence, both $\{x_1, x_2\}$ and $\emptyset$ reach a state containing True in no more than three steps. But $\{x_1\}$ and $\{x_2\}$ reach it in four steps, too many to obtain True as output. This example shows how the solutions generated achieve the goal of producing the correct output not by rote memorization of the inputs but by producing a complex interactions between the different reactions.

5 Conclusions

In this work a new biologically inspired evolutionary algorithm, called evolutionary reaction systems (EvoRS), has been defined. It is based on reaction systems, an expressive and powerful computational formalism inspired by chemical reactions, recently defined by Rozenberg and coworkers. We have shown that the performances of EvoRS are comparable, and in some cases even better, than the ones of other well known machine learning algorithms (including Bayesian methods, neural networks, support vector machines and standard genetic programming) on a set of case studies including real-life applications. This encourages us to pursue the study of EvoRS, with the objective of making it an established evolutionary algorithm. Future work will be focused in two directions. The first one is to study the best parameter settings for EvoRS and an assessment of its generalization ability. Since this method is new, an extensive study of the influence of different parameters is definitely needed. Another direction of research is the definition of a new version of EvoRS capable of handling continuous variables. This step would allow EvoRS to be applicable to a wider set of problems, including symbolic regression ones.

Acknowledgments. Leonardo Vanneschi gratefully acknowledges project PTDC/EIACCO/103363/2008 from Fundação para a Ciência e a Tecnologia, Portugal.

References

1. Caudill, M.: Neural networks primer, part i. AI Expert 2, 46–52 (1987)
2. Cristianini, N., Shawe-Taylor, J.: An introduction to support vector machines: and other kernel-based learning methods. Cambridge University Press (2000)
3. Ehrenfeucht, A., Rozenberg, G.: Basic Notions of Reaction Systems. In: Calude, C.S., Calude, E., Dinneen, M.J. (eds.) DLT 2004. LNCS, vol. 3340, pp. 27–29. Springer, Heidelberg (2004)
4. Ehrenfeucht, A., Rozenberg, G.: Reaction systems. Fundamenta Informaticae 75, 263–280 (2007)
5. Ehrenfeucht, A., Rozenberg, G.: Introducing time in reaction systems. Theoretical Computer Science 410, 310–322 (2009)
6. Fogel, D.B.: Evolving computer programs. In: Fogel, D.B. (ed.) Evolutionary Computation: The Fossil Record, ch. 5, pp. 143–144. MIT Press (1998)
7. Fogel, L.J., Owens, A.J., Walsh, M.J.: Artificial Intelligence through Simulated Evolution. John Wiley (1966)
8. Frank, A., Asuncion, A.: UCI machine learning repository (2010), http://www.ics.uci.edu/~mlearn/
9. Friedberg, R.M.: A learning machine: Part 1. IBM J. Research and Development 2(1), 2–13 (1958)
10. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. SIGKDD Explor. Newsl. 11, 10–18 (2009), http://www.cs.waikato.ac.nz/ml/weka/
11. Heckerman, D.: A tutorial on learning with bayesian networks. In: Innovations in Bayesian Networks. SCI, vol. 156, pp. 33–82. Springer, Heidelberg (2008)
12. Kantschik, W., Banzhaf, W.: Linear-Tree GP and Its Comparison with Other GP Structures. In: Miller, J., Tomassini, M., Lanzi, P.L., Ryan, C., Tetamanzi, A.G.B., Langdon, W.B. (eds.) EuroGP 2001. LNCS, vol. 2038, pp. 302–312. Springer, Heidelberg (2001)
13. Kantschik, W., Banzhaf, W.: Linear-Graph GP - A New GP Structure. In: Foster, J.A., Lutton, E., Miller, J., Ryan, C., Tettamanzi, A.G.B. (eds.) EuroGP 2002. LNCS, vol. 2278, pp. 83–92. Springer, Heidelberg (2002)
14. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
15. Miller, J.F., Thomson, P.: Cartesian Genetic Programming. In: Poli, R., Banzhaf, W., Langdon, W.B., Miller, J., Nordin, P., Fogarty, T.C. (eds.) EuroGP 2000. LNCS, vol. 1802, pp. 121–132. Springer, Heidelberg (2000)
16. Montana, D.J.: Strongly typed genetic programming. Evolutionary Computation 3(2), 199–230 (1995)
17. O'Neill, M., Ryan, C.: Grammatical evolution. IEEE Transactions on Evolutionary Computation 5(4), 349–358 (2001)
18. O'Neill, M., Vanneschi, L., Gustafson, S., Banzhaf, W.: Open issues in genetic programming. Genetic Programming and Evolvable Machines 11, 339–363 (2010)
19. Orr, M.J.L.: Introduction to radial basis function networks. Technical report, Centre For Cognitive Science, University of Edinburgh, Edinburgh (1996)
20. Platt, J.C.: A fast algorithm for training support vector machines. Technical report, Microsoft Research, Redmond, USA (1998)

21. Poli, R., Langdon, W.B., McPhee, N.F.: A field guide to genetic programming (2008), http://lulu.com, http://www.gp-field-guide.org.uk (With contributions by J. R. Koza)
22. Rish, I.: An empirical study of the naive bayes classifier. In: IJCAI 2001 Workshop on "Empirical Methods in AI" (2001)
23. Teller, A., Veloso, M.: PADO: A new learning architecture for object recognition. In: Ikeuchi, K., Veloso, M. (eds.) Symbolic Visual Learning, pp. 81–116. Oxford University Press (1996)
24. Whigham, P.A.: Grammatical Bias for Evolutionary Learning. PhD thesis, School of Computer Science, University College, University of New South Wales, Australian Defence Force Academy, Canberra, Australia (October 14, 1996)

Optimizing the Edge Weights in Optimal Assignment Methods for Virtual Screening with Particle Swarm Optimization

Lars Rosenbaum, Andreas Jahn, and Andreas Zell

University of Tuebingen, Center for Bioinformatics (ZBIT),
Sand 1, 72076 Tübingen, Germany
{lars.rosenbaum,andreas.jahn,andreas.zell}@uni-tuebingen.de

Abstract. Ligand-based virtual screening experiments are an important task in the early drug discovery stage. In such an experiment, a chemical database is searched for molecules with similar properties to a given query molecule. The optimal assignment approach for chemical graphs has proven to be a successful method for various cheminformatic tasks, such as virtual screening. The optimal assignment approach assumes all atoms of a query molecule to have the same importance. This assumption is not realistic in a virtual screening for ligands against a specific protein target. In this study, we propose an extension of the optimal assignment approach that allows for assigning different importance to the atoms of a query molecule by weighting the edges of the optimal assignment. Then, we show that particle swarm optimization is able to optimize these edge weights for optimal virtual screening performance. We compared the optimal assignment with optimized edge weights to the original version with equal weights on various benchmark data sets using sophisticated virtual screening performance metrics. The results show that the optimal assignment with optimized edge weights achieved a considerably better performance. Thus, the proposed extension in combination with particle swarm optimization is a valuable approach for ligand-based virtual screening experiments.

1 Introduction

The field of cheminformatics deals with in-silico approaches that are applied in the early stages of the drug discovery pipeline. The ranking of a chemical database with respect to a given query molecule, also known as ligand-based virtual screening (VS), represents one of the key tasks in cheminformatics [1,19]. The aim of a ligand-based VS experiment is to enrich molecules with similar properties (e.g. biological activity) to the query molecule in a preferably small top fraction of the ranked database. Generally, the database is sorted by a similarity function that measures the similarity between the query molecule and each database molecule. Database molecules with similar properties to the query molecule are assigned a top rank, whereas molecules with different properties are assigned a low rank. To assess the desired properties and to evaluate the

M. Giacobini, L. Vanneschi, and W.S. Bush (Eds.): EvoBIO 2012, LNCS 7246, pp. 26–37, 2012.

outcome of a VS run, the enriched molecules are then further analyzed by means of biological assays.

In recent years, a plethora of different similarity measures were proposed and the development of new functions is still a field of active research [2,9]. A common way to calculate the similarity between two molecules is to interpret the molecules as chemical graphs. The similarity between those graphs is then measured by means of graph kernels. Fröhlich et al. introduced the optimal assignment of chemical graphs to the field of cheminformatics [6,7]. The optimal assignment kernel and its extensions have proven to be a valuable similarity function for cheminformatic problems, such as VS [13,14].

The optimal assignment assumes every part of a query molecule to be equally important. However, this assumption is often not realistic in a VS for ligands that are biologically active against a specific protein target. Depending on the binding mode of a ligand, not all of its substructures are of the same importance. Parts of a ligand that exhibit important interactions (e.g. H-bonds) to the protein target should be more important than parts that do not directly interact with the protein.

The aim of this study is twofold. First, we propose an extension of the optimal assignment approach, which allows for assigning different importance to the atoms of a query molecule by weighting the edges of an optimal assignment. The importance of the atoms can be assigned by looking for potential interactions between the target and the ligand if a crystal structure and expert knowledge of the protein target is available. However, often crystal structures are not available or it is hard to interpret them. Hence, the second aspect of this study concerns the optimization of the edge weights of the optimal assignment. We show that the edge weights can be optimized with respect to an optimal VS performance on a data set of known actives and inactives. Using the VS performance on a data set as objective function results in a fitness landscape with multiple local optima. Evolutionary algorithms for numerical optimization problems are suited to optimize problems with such a fitness landscape. We employed particle swarm optimization (PSO) [16], a popular evolutionary algorithm that has been successfully applied in various practical tasks [8,25], to optimize the edge weights of the optimal assignment.

In the experiments, we compared the optimal assignment with optimized edge weights to the original version with equal edge weights. To assess the performance of both methods, we used sophisticated VS performance metrics that are able to measure the overall performance as well as the so called early enrichment of molecules, which is important in real-world VS experiments.

The results show that the optimization of edge weights considerably improved the overall VS performance as well as the early enrichment on various VS benchmark data sets. Additionally, the method is able to enrich molecules with a different scaffold, which is a desired behavior for real-world VS runs. Thus, we think that optimization of the edge weights of an optimal assignment is a valuable method if no crystal structure but a data set of known actives and inactives is available.

2 Methods

This section first introduces the optimal assignment kernel as similarity measure for molecular graphs and motivates the importance of different optimal assignment edge weights. Then, we present an evolutionary algorithm to optimize these edge weights. Finally, the experimental setup used to evaluate the performance of the optimal assignment with optimized edge weights is introduced.

2.1 Optimal Assignment Kernel

The optimal assignment kernel (OAK) [6,7] measures the similarity between two molecular graphs by finding an optimal mapping of the atoms of the smaller molecule on a subset of the atoms of the larger molecule. An optimal mapping of an atom on another atom results in an optimal assignment edge (Fig. 1). A mapping is called optimal if the mapping maximizes the pairwise sum of atom similarities. The optimal assignment is performed on the matrix S of pairwise, inter-molecule atom similarities. A pairwise atom similarity S_{ij} is calculated by a radial basis function (RBF) on the physio-chemical descriptors of each atom. The OAK uses 24 atom and 8 bond descriptors of the chemical expert system of JOELib2 [11].

From a chemical perspective, the local environment of an atom influences its chemical properties. Hence, the OAK includes information of the topological neighbors and the bonds up to a predefined depth in the atom similarity calculation. The information is gathered by a recursive atom-wise similarity calculation on the neighbors. The recursive atom-wise similarities are weighted by a decay parameter because the influence of neighboring atoms on the chemical properties of an atom decreases with increased topological distance.

Given two molecular graphs A and B with atoms $a_1, \ldots, a_m$ and $b_1, \ldots, b_n$ and the matrix S of pairwise atom similarities, the optimal assignment problem can be formulated as finding an optimal permutation π of indices that maximizes the objective function of Equation 1.

$$S(A,B) = \begin{cases} \max_\pi \sum_{i=1}^{m} S_{i\pi(i)} & \text{if} \quad n > m \\ \max_\pi \sum_{j=1}^{n} S_{\pi(j)j} & \text{otherwise} \end{cases} \tag{1}$$

Using the Hungarian method [21], an optimal solution for the optimal assignment problem can be computed in $O(\max(m,n)^3)$.

The sum of all pairwise atom similarities in Equation 1 increases with the number of atoms that a molecule contains. To obtain comparable similarity values for molecules of arbitrary sizes the result $S(A,B)$ of the optimal assignment is normalized to a range of $[0,1]$ by Equation 2.

$$S_{OA}(A,B) = \frac{S(A,B)}{\sqrt{S(A,A)S(B,B)}} \tag{2}$$

An example of an optimal assignment of two molecular graphs is visualized in Fig. 1.

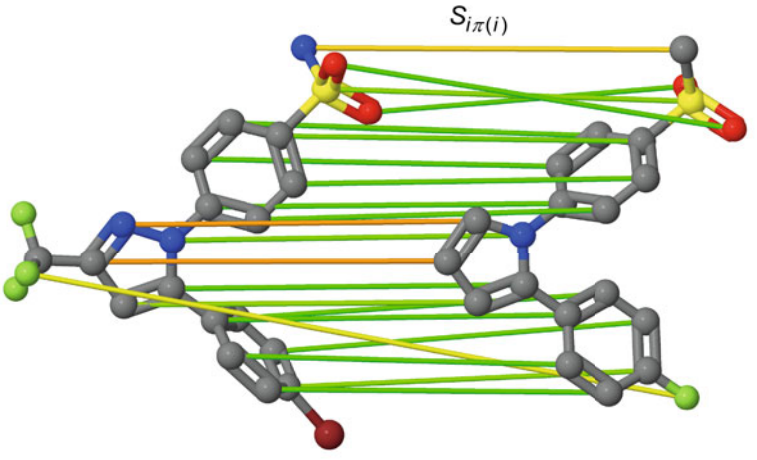


Fig. 1. Optimal assignment of two molecules of the COX2 dataset. The atom assignments are based on pairwise-atom similarities calculations of the OAK. Green optimal assignment edges represent a high atom similarity whereas red edges indicate a low atom similarity.

Optimal Assignment Edge Weights. In a VS for ligands against a specific protein target, a single active query molecule is used to search a chemical database for other biologically active compounds. Generally, not all parts of the query molecule have the same importance for activity. For instance, the exact topology of a substructure that is crucial for the molecule's binding to the protein target is usually more important than the topology of some linker region. Hence, important substructures should receive more attention in the optimal assignment than unimportant substructures.

A different importance can be assigned to the atoms of a query molecule by weighting the optimal assignment edges that originate from the atoms. Edges that are part of more important substructures receive larger weights whereas edges within less important parts of the query molecule receive smaller weights. Consequently, the contribution of an assignment to the sum of pairwise atom similarities increases with the importance of the substructure that contains the assignment.

Assuming a fixed query molecule Q with atoms $q_1, \ldots, q_m$, the objective function of the optimal assignment problem (Equation 1) is modified to Equation 3.

$$S(Q,B) = \begin{cases} \max_\pi \sum_{i=1}^{m} w_i S_{i\pi(i)} & \text{if} \quad n > m \\ \max_\pi \sum_{j=1}^{n} w_{\pi(j)} S_{\pi(j)j} & \text{otherwise} \end{cases} \tag{3}$$

The optimal assignment edge weights w_i represent the importance of the corresponding atom of the query molecule. The edge weights are fixed throughout the similarity calculations of a VS run and need to be determined prior to the VS of a chemical database.

2.2 Edge Weight Optimization

A possible approach to assign edge weights would be to look at the binding mode of the query molecule to the protein target. However, depending on the availability of crystal structures and literature, the exact binding mode is often unknown. Hence, it is often unknown which parts of a query molecule are important for activity and which parts are less important. However, a data set of known ligands and decoys is usually available. The information of this data set can be used to optimize the edge weights of a query molecule for optimal VS performance.

We employed evolutionary algorithms to optimize the edge weights of a given query structure. The VS performance on a data set of known ligands and decoys was used as fitness function for the optimization. Changing the edge weights can dramatically change the actual optimal assignment. This fact results in a fitness landscape with multiple local optima. Thus, an evolutionary algorithm for numerical optimization with a good exploratory behavior should be used for optimization. Differential evolution (DE) [23] and PSO are popular evolutionary algorithms for tackling numerical problems with multiple local optima or dynamically changing fitness functions. Implementations of the two algorithms are available in the optimization framework EvA2 [20].

In preliminary experiments (not published), we compared DE and PSO using the EvA2 standard parameters with respect to convergence speed and model quality. PSO and DE resulted in a similar model quality but PSO needed on average 1,000 iterations less to converge, which is a considerably faster convergence speed. Thus, we decided to employ PSO to optimize the edge weights of the optimal assignment. Additionally, several multi-run PSO optimizations resulted in similar fitness values indicating that multi-runs might not be necessary for optimizing the edge weights of an optimal assignment.

Particle Swarm Optimization. PSO is a population based optimization technique inspired by swarms of fish or birds. Each individual, or swarm particle, x_i is characterized by its position in the problem space and its current travel velocity $v_i(t)$ that allows it to move in the problem space. The particles are arranged in a logical topology which defines a neighborhood $N(x_i)$ for each particle x_i. The motion of a particle is influenced by both individual knowledge and knowledge of neighboring swarm individuals.

At iteration t, a swarm particle x_i is attracted by the best position x_i^h in the particle's history and the best position x_i^n found by its neighboring particles' history, resulting in Equations 4 and 5.

$$v_i(t+1) = \chi\left[v_i(t) + r_1\phi_1(x_i^p - x_i(t)) + r_2\phi_2(x_i^n - x_i(t))\right] \tag{4}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{5}$$

The parameters ϕ_1 and ϕ_2 control the trade-off between the different attractors and r_1 and r_2 are uniform random samples for randomized exploration. The constriction approach includes a constriction factor χ to assure that the swarm reliably converges [5]. Using the constriction approach, it is not necessary to limit

the velocity vector to a maximum velocity v_{max}. Commonly used neighborhoods are ring, 2D-grid, or star topologies [17].

2.3 Experimental Setup

Data Sets. All VS experiments were performed using a subset of the Directory of Useful Decoys (DUD) Release 2 [12]. The DUD contains known actives and mimetic decoys for 40 protein targets. We used the same 13 subsets of the DUD as in an earlier VS study [13] because these data sets contain a sufficient number of different scaffolds. On a data set with a low number of scaffolds the VS performance would be mainly based upon a trivial enrichment. An overview of the data sets and their corresponding protein targets can be found in Table 1.

The DUD was designed to serve as an unbiased, publicly available benchmark database for evaluation of docking methods. In contrast to docking methods, ligand-based VS methods require a biologically active query molecule. In line with other VS studies on the DUD data sets [4,13], we used the ligands of the complexed crystal structures that were used to identify the binding sites for docking algorithms as query molecules.

Table 1. DUD data sets for VS experiments

data set	target protein	actives	decoys
ACE	Angiotensine-converting enzyme	49	1797
ACHE	Acetylcholinesterase	107	3892
CDK2	Cyclin-dependent kinase	72	2074
COX2	Cyclooxygenase-2	426	13289
EGFR	Epidermal growth factor receptor	475	15996
FXA	Factor Xa	146	5745
HIVRT	HIV reverse transcriptase	43	1519
INHA	Enoyl ACP reductase	86	3266
P38	P38 mitogen activated protein	454	9141
PDE5	Phosphodiesterase 5	88	1978
PDGFRB	Platelet derived growth factor receptor kinase	170	5980
SRC	Tyrosine kinase	159	6319
VEGFR2	Vascular endothelial growth factor receptor	88	2906

Virtual Screening Metrics. In recent years, a plethora of sophisticated evaluation metrics for VS experiments have been suggested [18]. According to recommendations for the evaluation of VS experiments [15], the VS results should be analyzed under at least two different aspects.

First, the overall performance on the complete data set. The overall performance can be measured by the well known area under the ROC curve (AUC). The ROC curve plots the fraction of correctly predicted actives (true positive rate) against the fraction of inactives that are falsely predicted as active (false positive rate). The AUC can achieve values in the interval $[0, 1]$. The higher the AUC value, the better the performance.

The second aspect is the early enrichment performance, which originates from real-world screening applications. In those screenings only the top ranked molecules are further evaluated in biological assays because of cost and time requirements. We employed two different metrics to measure the early enrichment performance. For both metrics, increased values indicate an increased early enrichment.

The first early enrichment metric, the BEDROC score [24], extends the AUC by a decreasing exponential weighting function that reduces the influence of lower ranked structures. The influence of the early enrichment on the final BEDROC score is controlled by a parameter α. High values of α increase the importance of top ranked structures. We used $\alpha = 53.6$, which means that 80% of the final BEDROC score is based on the performance in the first 3% of the ranked data set. Like the AUC, the BEDROC score adopts values in the range $[0, 1]$.

Another metric that is commonly used for the early enrichment problem is the enrichment factor (EF). The EF measures the enrichment of actives at a predefined fraction of the data set ($x\%$) as defined in Equation 6. In particular, the EF represents how much more enrichment could be found compared to a random enrichment of actives. In contrast to the other two metrics, the EF is not bounded to the interval $[0, 1]$.

$$EF_{@x\%} = \frac{N_{\text{actives seen}}}{N_{@x\%}} / \frac{N_{\text{actives}}}{N_{\text{actives}} + N_{\text{decoys}}} \tag{6}$$

Evaluation Setup. To robustly evaluate the equally weighted and the edge weight optimized OAK, we generated 25 randomized 50/50 splits. The first half of a data set was used to optimize the optimal assignment edge weights for optimal AUC performance. In each iteration of the PSO, a complete VS run was performed and evaluated on the first half of the data set. The second half of a data set was used as external test set to obtain 25 unbiased VS results for the calculation of mean and standard deviation.

We used the constriction PSO implementation of Eva2 with default parameters: $\phi_1 = \phi_2 = 2.05$, $\chi \approx 0.73$, an initial velocity $v = 0.2$, and a population size of 30 individuals. These parameter settings worked well in an earlier problem solved with PSO. As neighborhood, we used a 2D-grid with range two. As preliminary experiments indicated that multi-runs result in similar fitness values, we performed only a single optimization run for each of the 25 randomized splits. Generally, multi-runs are necessary to draw statistically relevant conclusions for a given optimization problem. However, we think that a single run for each of the 25 different splits, which results in 25 optimizations in total, is enough to draw relevant conclusions.

3 Results

The goal of this section is to compare the overall performance as well as the early enrichment performance of the OAK with PSO optimized edge weights (PSO-OAK) to the original version with equal edge weights (OAK).

Table 2 shows the AUC performance and the BEDROC scores on the 13 employed benchmark data sets. The results are based on the averaged performance and its standard deviation on the external sets of the 25 randomized splits. We performed a standard paired t-test to test for significant differences between the performance values.

Table 2. AUC and BEDROC performance for equally weighted and PSO weight optimized OAK. **Bold** values indicate the best result with respect to the metric.

	OAK		PSO-OAK	
data set	AUC	BEDROC	AUC	BEDROC
ACE	0.717 ± 0.046	0.438 ± 0.077	**0.934 ± 0.023**	**0.602 ± 0.090**
ACHE	0.666 ± 0.027	0.397 ± 0.050	**0.892 ± 0.027**	**0.693 ± 0.051**
CDK2	0.542 ± 0.043	0.196 ± 0.054	**0.809 ± 0.023**	**0.471 ± 0.053**
COX2	0.913 ± 0.008	0.795 ± 0.013	**0.972 ± 0.007**	**0.930 ± 0.012**
EGFR	0.895 ± 0.006	0.592 ± 0.023	**0.968 ± 0.003**	**0.805 ± 0.015**
FXA	0.462 ± 0.020	0.053 ± 0.029	**0.822 ± 0.026**	**0.416 ± 0.051**
HIVRT	0.566 ± 0.057	**0.262 ± 0.078**	**0.725 ± 0.045**	0.194 ± 0.101
INHA	0.615 ± 0.042	0.653 ± 0.038	**0.939 ± 0.016**	**0.773 ± 0.040**
P38	0.403 ± 0.016	0.134 ± 0.023	**0.763 ± 0.013**	**0.271 ± 0.039**
PDE5	0.610 ± 0.033	0.504 ± 0.062	**0.826 ± 0.024**	**0.632 ± 0.052**
PDGFRB	0.495 ± 0.034	0.252 ± 0.041	**0.833 ± 0.022**	**0.627 ± 0.041**
SRC	0.662 ± 0.030	0.371 ± 0.367	**0.893 ± 0.018**	**0.623 ± 0.045**
VEGFR2	0.250 ± 0.036	0.058 ± 0.030	**0.754 ± 0.033**	**0.193 ± 0.071**

The PSO-OAK outperformed the OAK on all data sets with respect to AUC performance, which means that the PSO-OAK showed a considerably better overall performance. The AUC performance benefit ranged from 0.059 on the COX2 data set up to 0.504 on the VEGFR2 data set. On 6 out of the 13 benchmark data sets, the OAK performed with an AUC between 0.250 and 0.566 close to or worse than random. In contrast, the PSO optimized OAK exhibited an AUC performance from 0.725 to 0.833 on these 6 data sets, which is considerably better than random.

The PSO-OAK achieved an improved BEDROC score on all data sets but the HIVRT data set, which means that the enrichment of biologically active structures in roughly the first 3% of the data set could be increased on all but one data set. The performance loss on the HIVRT data set was 0.068. On the other 12 data sets, the BEDROC score was improved by 0.120 for INHA up to 0.375 for PDGFRB.

Table 3 shows the EF at 1% and 5% of the data set and represents the early enrichment performance at two small fractions of the complete data set. At a fraction of 1% the PSO-OAK yielded a better enrichment on 11 of the 13 benchmark data sets. On these data sets, the improvement was between 3.15 on the PDE5 data set and 16.60 on the PDGFRB data set. As for the BEDROC score, the enrichment of the PSO-OAK was, with a decrease of 5.78, significantly worse on the HIVRT data set. On the COX2 data set, the OAK and the PSO-OAK achieved a comparable enrichment.

Table 3. Enrichment factors (EFs) at 1% and 5% of the data set for equally weighted and PSO weight optimized OAK. **Bold** values indicate the best result with respect to the metric whereas *italics* indicate results that are statistically indistinguishable.

	OAK		PSO-OAK	
data set	$EF_{@1\%}$	$EF_{@5\%}$	$EF_{@1\%}$	$EF_{@5\%}$
ACE	23.281 ± 5.367	7.613 ± 1.305	**28.437 ± 7.068**	**11.681 ± 2.045**
ACHE	23.112 ± 3.228	5.964 ± 0.801	**28.291 ± 3.388**	**13.945 ± 1.173**
CDK2	7.260 ± 3.233	3.152 ± 0.628	**20.394 ± 3.233**	**7.457 ± 0.935**
COX2	*30.018 ± 0.376*	14.060 ± 0.425	*30.624 ± 0.550*	**17.690 ± 0.278**
EGFR	25.730 ± 1.241	10.457 ± 0.471	**30.026 ± 0.844**	**15.161 ± 0.438**
FXA	3.093 ± 1.817	0.610 ± 0.358	**15.683 ± 3.159**	**10.127 ± 1.294**
HIVRT	**15.077 ± 5.358**	*3.633 ± 1.164*	9.294 ± 5.577	*3.670 ± 1.616*
INHA	33.799 ± 3.009	10.798 ± 0.825	**37.407 ± 1.241**	**14.312 ± 1.133**
P38	4.026 ± 0.900	1.476 ± 0.218	**6.751 ± 1.575**	**3.999 ± 0.356**
PDE5	19.151 ± 3.187	5.483 ± 0.815	**22.297 ± 0.622**	**7.792 ± 1.000**
PDGFRB	12.617 ± 2.735	4.120 ± 0.604	**29.220 ± 2.343**	**10.898 ± 0.640**
SRC	15.495 ± 2.164	7.858 ± 0.798	**27.641 ± 3.667**	**13.054 ± 0.910**
VEGFR2	2.264 ± 1.594	0.875 ± 0.397	**7.454 ± 4.895**	**3.963 ± 1.413**

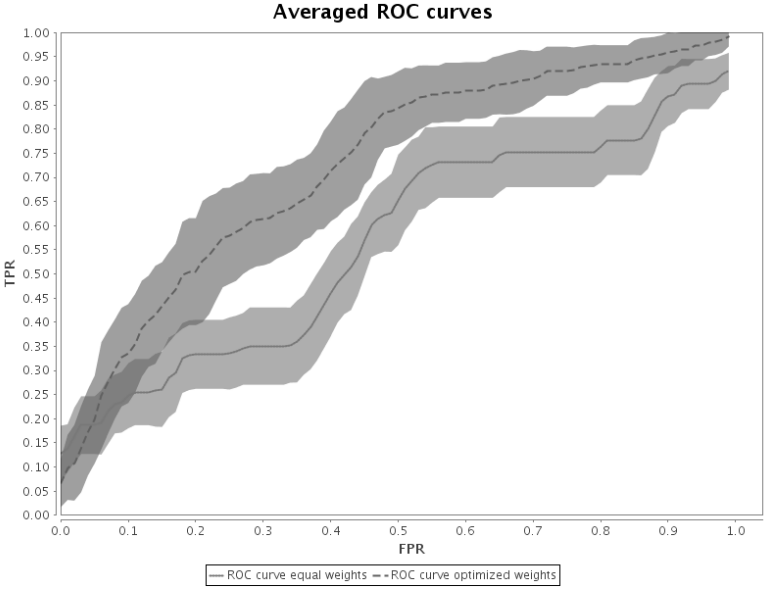


Fig. 2. Receiver operator characteristics (ROC) on the HIVRT data set for equally weighted and PSO weight optimized OAK. The standard deviations of the ROC curves are indicated by shaded tubes.

At a fraction of 5% of the data set, the PSO-OAK also achieved a considerably better enrichment on the COX2 data set. Thus, the optimization of edge weights improved the enrichment at 5% on 12 out of 13 data sets. The increase on these data sets ranged from 2.31 for PDE5 up to 9.52 for FXA. On the HIVRT data set, the OAK and the PSO-OAK showed a comparable performance.

The performance values of the two methods on the HIVRT data set exhibit an interesting behavior. While the overall performance of the PSO-OAK was considerably better, the early enrichment up to 3% of the data set was worse compared to the OAK. To investigate these performance differences on the HIVRT data set in more detail, the ROC curves of the OAK and the PSO-OAK are depicted in Figure 2. The curves show that the enrichment of true positives of the OAK was better up to a false positive rate of 4%, which accounts for the significantly better BEDROC and $EF_{@1\%}$ performance. At larger fractions of false positives the PSO-OAK enriched more true positives, which underlines the good overall performance. Another aspect concerns the different course of the curves of the two methods. The PSO-OAK shows a smooth curve progression, whereas the OAK exhibits a nearly stepwise increase.

4 Discussion and Conclusion

In this study we proposed an extension of the optimal assignment approach for chemical graphs that uses PSO to optimize the edge weights for optimal VS performance. An improved in-silico VS performance, in particular an increased early enrichment, helps to increase the success rate of further biological assays and thus, reduces cost and time requirements.

The optimization of edge weights of the optimal assignment substantially increased the overall VS performance and the early enrichment on various benchmark data sets. The median improvements amount to 35% for the AUC, 68% for the BEDROC score, 22% for the EF at 1% of the data set, and 66% for the EF at 5% of the data set. This considerable performance gain can be explained by the additional information the edge weight optimized OAK receives from the optimization step. Recent studies showed that the importance of substructures of a molecule can be extracted from an accurate machine learning model [3,22]. The same works the other way around. Optimizing the edge weights, which represent the importance of atoms, for optimal VS performance on a problem-specific data set is similar to building a machine learning model on such a data set. Thus, the proposed extension of the OAK includes a model of activity against the specific protein target, which is not the case for the original optimal assignment approach. Consequently, a considerable performance gain is to be expected and the performance could be similar to model-based approaches. The comparison of the OAK with optimized edge weights to model-based approaches should be addressed in further studies. A drawback compared to the equally weighted OAK is that the optimization of edge weights can lead to substantial overfitting, a common problem of machine learning approaches. However, overfitting could not be observed on the employed benchmark data sets because the performance on the external test sets was promising.

The ROC curves on the HIVRT reveal another possible benefit of the edge weight optimized OAK. The nearly stepwise increase for the equally weighted OAK indicates that the approach accumulates numerous molecules with the same scaffold before discovering a new scaffold. As a specific scaffold contains both actives and inactives, the scaffold-wise accumulation leads to a nearly stepwise increase of the ROC curve. In contrast, the smooth curve progression of the edge weight optimized OAK indicates that the method accumulates molecules of different scaffolds. This capability to enrich molecules with different scaffolds, also called "scaffold-hopping", is important for a pharmaceutical company. Only substances with a substantially different scaffold compared to existing treatments can be patented [10].

To conclude, the OAK with optimized edge weights is a valuable method to improve the VS performance if a problem-specific data set is available. A further advantage is that the edge weights can be used to visualize the importance of the atoms of a query structure. In combination with structure-based approaches such a visualization could help to gain a better understanding of the binding mode of a ligand.

References

1. Bajorath, J.: Integration of virtual and high-throughput screening. Nat. Rev. Drug Discov. 1(11), 882–894 (2002)
2. Bender, A., Jenkins, J.L., Scheiber, J., Sukuru, S.C., Glick, M., Davies, J.W.: How Similar Are Similarity Searching Methods? A Principal Component Analysis of Molecular Descriptor Space. Chem. Inf. Model. 49(1), 108–119 (2009)
3. Bender, A., Mussa, H.Y., Glen, R.C., Reiling, S.: Molecular similarity searching using atom environments, information-based feature selection, and a nave bayesian classifier. Chem. Inf. Comput. Sci. 44(1), 170–178 (2004)
4. Cheeseright, T.J., Mackey, M.D., Melville, J.L., Vinter, J.G.: FieldScreen: Virtual Screening Using Molecular Fields. Application to the DUD Data Set. Chem. Inf. Model. 48(11), 2108–2117 (2008)
5. Clerc, M., Kennedy, J.: The particle swarm - explosion, stability, and convergence in a multidimensional complex space. IEEE T Evolut. Comput. 6(1), 58–73 (2002)
6. Fröhlich, H., Wegner, J.K., Sieker, F., Zell, A.: Optimal assignment kernels for attributed molecular graphs. In: ICML 2005: Proceedings of the 22nd International Conference on Machine Learning, pp. 225–232. ACM, New York (2005)
7. Fröhlich, H., Wegner, J.K., Sieker, F., Zell, A.: Kernel Functions for Attributed Molecular Graphs - A New Similarity-Based Approach to ADME Prediction in Classification and Regression. QSAR Comb. Sci. 25(4), 317–326 (2006)
8. Geis, M., Middendorf, M.: A Particle Swarm Optimizer for Finding Minimum Free Energy RNA Secondary Structures. In: Proc. IEEE Swarm Intelligence Symp., SIS 2007, pp. 1–8 (2007)
9. Geppert, H., Vogt, M., Bajorath, J.: Current trends in ligand-based virtual screening: Molecular representations, data mining methods, new application areas, and performance evaluation. Chem. Inf. Model. 50(2), 205–216 (2010)
10. Good, A.C., Hermsmeier, M.A., Hindle, S.: Measuring CAMD Technique Performance: A Virtual Screening Case Study in the Design of Validation Experiments. Comput.-Aided Mol. Des. 18(7), 529–536 (2004)

11. Guha, R., Howard, M.T., Hutchison, G.R., Murray-Rust, P., Rzepa, H., Steinbeck, C., Wegner, J., Willighagen, E.L.: The Blue Obelisk Interoperability in Chemical Informatics. Chem. Inf. Model. 46(3), 991–998 (2006)
12. Huang, N., Shoichet, B.K., Irwin, J.J.: Benchmarking Sets for Molecular Docking. Med. Chem. 49(23), 6789–6801 (2006)
13. Jahn, A., Hinselmann, G., Fechner, N., Zell, A.: Optimal assignment methods for ligand-based virtual screening. Cheminf. 1, 14 (2009)
14. Jahn, A., Rosenbaum, L., Hinselmann, G., Zell, A.: 4d flexible atom-pairs: An efficient probabilistic conformational space comparison for ligand-based virtual screening. Cheminform. 3(1), 23 (2011)
15. Jain, A.N., Nicholls, A.: Recommendations for Evaluation of Computational Methods. Comput.-Aided Mol. Des. 22(3-4), 133–139 (2008)
16. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948. IEEE Computer Society, Perth (1995)
17. Kennedy, J., Mendes, R.: Population structure and particle swarm performance. In: Proc. Congress Evolutionary Computation CEC 2002, vol. 2, pp. 1671–1676 (2002)
18. Kirchmair, J., Markt, P., Distinto, S., Wolber, G., Langer, T.: Evaluation of the Performance of 3D Virtual Screening Protocols: RMSD Comparisons, Enrichment Assessments, and Decoy Selection - What can We Learn from Earlier Mistakes? Comput.-Aided Mol. Des. 22(3-4), 213–228 (2008)
19. von Korff, M., Freyss, J., Sander, T.: Flexophore, a New Versatile 3D Pharmacophore Descriptor That Considers Molecular Flexibility. Chem. Inf. Model. 48(4), 797–810 (2008)
20. Kronfeld, M., Planatscher, H., Zell, A.: The EvA2 Optimization Framework. In: Blum, C., Battiti, R. (eds.) LION IV. LNCS, vol. 6073, pp. 247–250. Springer, Heidelberg (2010)
21. Kuhn, H.W.: The hungarian method for the assignment problem. Naval Res. Logist. 2, 83–97 (1955)
22. Rosenbaum, L., Hinselmann, G., Jahn, A., Zell, A.: Interpreting linear support vector machine models with heat map atom and bond coloring. Cheminf. 3(11) (2011)
23. Storn, R., Price, K.: Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. Tech. Rep. TR-95-012, ICSI (1995)
24. Truchon, J.F., Bayly, C.I.: Evaluating Virtual Screening Methods: Good and Bad Metrics for the "Early Recognition" Problem. Chem. Inf. Model. 47(2), 488–508 (2007)
25. Vahdat, A., NourAshrafoddin, N., Ghidary, S.: Mobile robot global localization using differential evolution and particle swarm optimization. In: Srinivasan, D., Wang, L. (eds.) 2007 IEEE Congress on Evolutionary Computation, pp. 1527–1534. IEEE Press, Singapore (2007)

Lévy-Flight Genetic Programming: Towards a New Mutation Paradigm*

Christian Darabos[1], Mario Giacobini[2,3], Ting Hu[1], and Jason H. Moore[1]

[1] Computational Genetics Laboratory, Dartmouth Medical School, Dartmouth College, Hanover, NH 03755, USA
[2] Computational Biology Unit, Molecular Biotechnology Center, University of Torino, Italy
[3] Department of Animal Production, Epidemiology and Ecology Faculty of Veterinary Medicine, University of Torino, Italy

Abstract. Lévy flights are a class of random walks inspired directly by observing animal foraging habits, in which the stride length is drawn from a power-law distribution. This implies that the vast majority of the strides will be short. However, on rare occasions, the stride are gigantic. We use this technique to self-adapt the mutation rate used in Linear Genetic Programming. We apply this original approach to three different classes of problems: Boolean regression, quadratic polynomial regression, and surface reconstruction. We find that in all cases, our method outperforms the generic, commonly used constant mutation rate of 1 over the size of the genotype. We compare different common values of the power-law exponent to the regular spectrum of constant values used habitually. We conclude that our novel method is a viable alternative to constant mutation rate, especially because it tends to reduce the number of parameters of genetic programing.

1 Introduction

To men, nature has always been a source of inspiration for innovation. This is true for all realms of technological advances. Since men first saw a bird, mankind wanted to fly and plane. Directly mimicking biological organisms has given rise to anodyne inventions such as velcro, and entire new fields of science, such as artificial intelligence. Evolutionary algorithms (EAs), including genetic programming (GP), are prime examples of Darwinian evolution concepts used to "intelligently" explore the solution space of problems too vast to enumerate exhaustively.

Although successfully applied to real-life problems, EAs generally suffer from a severe drawback: the number of parameters needing optimization before the EA is able to perform. Parameter setting and the necessity of tuning them to the specific problem is both time and resources consuming. And the number of possible combinations of parameters grows exponentially. GP, or in our case, linear genetic programming (LGP), is no exception to the rule. The maximum number of generations, the mutation rate, the type of selection, or the size of the genotype are only a small subset of all possible variables.

In this work, we take another page from nature and apply a biological concept to make an attempt at optimizing LGP systems. Instead of fixing a single global rate, or

* Authors contributions are all equal. Names appear in alphabetical order.

M. Giacobini, L. Vanneschi, and W.S. Bush (Eds.): EvoBIO 2012, LNCS 7246, pp. 38–49, 2012.

using a complicated variable rate function, we use the Lévy flight paradigm, a particular case of random walk, to draw the mutation rate from a heavy-tailed distribution. In Section 2 we give a detailed description of Lévy systems and of linear genetic programming, we then proceed to describe our methods in Section 3. In Section 4, we describe, analyze, and discuss our simulation results. Finally, we draw some conclusions and offer possible future research direction in Section 5.

2 Background

2.1 Lévy Walks and Flights as Optimal Search Strategies

Until 1995 [5,20] animal movement was mainly modeled by random walks (RWs) [2,3]. A RW is a stochastic process where the location of a point in a space varies in time according to a defined set of probabilistic rules. The hypothesis that animal foraging behavior could be better described by a particular class of RWs, Lévy dynamics (LDs), was first proposed in [16]. According to this paradigm, the distance (step length) travelled by a point between reorientation events is described by a probability distribution that is heavy- tailed, i.e. without finite variance, usually power-law or Pareto distributions. Here, the probability density function of a step length $x \in [x_{min}, \infty)$ is draw from

$$P(x) = Cx^{-\gamma}$$

where the γ satisfies $1 < \gamma \leq 3$. Exponents equals to 1 do not correspond to a well-defined probability distribution, while exponents greater than 3 correspond to distributions with finite variance. An important feature of LDs is that they are scale-free: they do not have any characteristic spatial scale, exhibiting the same patterns regardless of the range over which they are viewed.

These distributions do not satisfy the central limit theorem's hypothesis, therefore standard results on the long-term limit of RWs do not apply for LDs. Instead, they are superdiffusive, the long-term mean-squared displacement of the point is proportional to the time from the beginning of the process raised to a given exponent strictly greater than 1.

When the time taken to complete a given step is somehow proportional to its length, the term Lévy walk (LW) is used. Otherwise, when the the movement of the point is instantaneous, Lévy flight (LF) is preferred, even if these two terms are often used as synonyms when referred to animal moving behaviors. Taking into considerations all the above observations, LWs and LFs provide an interesting paradigm that allows for a continuous transition between different movements: from ballistic (straight-line) motion ($\gamma = 1$) to diffusive (Brownian) RWs ($\gamma \geq 3$), passing through superdiffusion (when $1 < \gamma \leq 3$).

The use of LW in foraging behavior of animals was initiated by empirical papers that demonstrated the presence of a heavy-tailed distribution in data describing the movements of fruit flies [5] and wandering albatrosses [20]. The success on the use of these paradigms is surely also due to the theoretical studies of the efficiency of a point carrying out a random walk search with a power-law distribution of its movements in an environment designed to model patchily distributed search targets [19,21]. These studies

showed that LWs are more efficient than non-Lévy walks and the optimal Lévy exponent is approximately 2. In fact, from the observation that diffusive movements tend to repeatedly search the same space, while ballistic movements are less suited to exploiting the patchy nature of the food environment, Viswanathan *et al.* [19,20,21] postulated that a LW with exponent around the value 2 represents an optimal compromise between the Brownian and the ballistic search modes.

The studies of Viswanathan *et al.* initiated a great interest in LWs, both in the analysis of empirical papers (a large number of animal species behaviors were studied, from reindeer, to spider monkeys, from gray seals to bees, moths and marine predators) and in the theoretical study of the model and it generalization to different fields. Recently, a re-analysis by Edwards *et al.* [7] of the original data showed flaws in the statistical methods used to analyze them. A recent re-analysis of previously published statistical studies overwhelmingly rejected the original Lévy model for almost all datasets tested [8].

Further theoretical work showed that alternative search strategies can outperform LWs [10]. For example, they showed that *if there is a small increase in the initial distance between forager and target, or a short period of time following detection for which a target is available for future searches, the optimal Lévy exponent decreases from 2 towards the ballistic limit of 1. Furthermore, the efficiency of an LW relative to that of a ballistic search is greatly reduced. Therefore, the theoretical optimum of a 2 LW is not as robust as is widely thought.*

However, it is quite well established that a wide range of movement strategies can lead to the observation of heavy-tailed patterns. Therefore, *one of the key questions in the field of optimal foraging is: under what circumstances is it advantageous for a forager to follow a movement strategy based on a LW? Crucially, the answer to this question depends on what alternative strategies are realistically available to the forager* [10]. Even the exact modelization of animal foraging behavior is still under discussion and strongly depends on the assumption of the model, the Lévy-walk and -flight paradigms interestingly apply in many situations, suggesting a new metaphor for designing robust and efficient heuristic search strategies.

2.2 Genetic Programming and Linear Genetic Programming

Genetic Programming (GP) is an inductive learning technique in which a population of computer programs evolve using Darwinian principals towards an optimal solution to a predefined problem [1,12]. In traditional GP, individuals are programs in the form of trees where nodes are operators and terminals are variables or constant values. For a finite number of successive generations, individuals are selected for the adequacy to the problem at hand: their *fitness* is evaluated. Operators such as recombination or mutation are used on selected *parent* programs in order to produce offspring that are different and possibly closer to generate a globally optimal solution. If their fitness is higher than that of their parents, they may replace them in the next generation. There are countless different methods for selecting, mutating, recombining, and replacing individual solutions, making the parameter space of GP extremely large. Tree-based GP also suffers what is called the *bloat*, where the trees are becoming increasingly deep and unbalanced, dramatically reducing the efficiency of genetic operators [14,17].

In evolutionary algorithms, including GP and evolution strategy, a broad range of (self-)adaptive mutation rates were proposed. These can decrease linearly following a simulated annealing technique [9], or be much more elaborated, including sophisticated statistical frameworks [18]. In our case, we are comparing our results to the most common implementation: the constant mutation rate.

To remedy this problem, in this work we will consider Linear Genetic Programming (LGP) [4]. In LGP, programs are now sets of linear instructions. LGP is very similar to a computing machine system composed of a set of registers and instructions that operate upon their content. In the linear representation, an individual is an imperative program denoting a sequence of instructions that are executed sequentially from top to bottom. Each instruction includes an operation, one or multiple operands, and a *return* that the result of that operation is assigned to. Both operands and return are stored in registers with varying read/write permissions. In general, input registers (e.g. $\mathtt{r_x}, \mathtt{r_y}$) hold the program inputs and can only be read and serve as operand. Additional calculation registers can be read and written and can be used as either operand or return. Calculation registers are usually initialized with a constant value, (*e.g. 1*) for numeric search problem and `FALSE` for Boolean programs. One of the calculation registers (usually $\mathtt{r_0}$) is assigned as the output register that after a program is executed, the value stored in $\mathtt{r_0}$ will be returned as the output.

3 LGP with Lévy-Flight Mutation

In order to streamline the search for an optimal mutation rate, we use a Lévy-flight approach implemented into Linear Genetic Programming (LGP) system. Similar strategies were applied in other other subfields of EA, notably Evolutionary Programming [13]. In LGP, point mutation can be applied to any loci, that is, a return register can be replaced, and so can an operator or an operand. These mutations occur with a probability p_m, or at a fixed rate, which is usually a function of the number of loci N, generally x/N, where $x << N$. This will become the *standard* against which we will compare our Lévy-flight implementation.

In this preliminary study, we focus on a mutation-only based search framework (i.e. no recombination will be used) and we will predefine a common fixed length for all LGP programs. More specifically, all individuals/programs will have exactly $N = I \times 4$ loci if we constrain a program with I instructions. Each instruction is of the following form:

$$\mathtt{r_{ret}} = \mathtt{r_{opr_1}}\ \ \mathtt{operator}\ \ \mathtt{r_{opr_2}}$$

where $\mathtt{r_{ret}}$ is chosen from the calculation register set, $\mathtt{r_{opt}}$ can be either calculation or input register, and `operator` is one of the possible operations.

The mutation-based Lévy-flight LGP algorithm starts with randomly generating a population of a given size $|P|$. The configuration of population size and other parameters will be detailed in Section 4. Next, the fitness of each individual in this initial population is evaluated. Then the evolution process enters a generational iteration outlined as follows.

1. *Mutate each individual using a mutation rate m drawn from a power-law distribution* $p(l) = Cl^{-\gamma}$*, where N is the fixed length of an individual,* $l = 1,2,3,...,N$*, and the constant* $C = 1/\sum l^{-\gamma}$*;*
2. *Evaluate offspring;*
3. *Choose by tournament selection the next generation from the competition pool that consists of* $|P|$ *parent and* $|P|$ *offspring individuals;*
4. *Go to Step 1 if termination criterion is not met.*

In order to obtain statistically significant results, we repeat each experiment between 1000 and 10,000 times in 3 different types of problems.

4 Experimental Results

The performance of our Lévy-flight GP is evaluated by comparing to conventional fixed-rate mutation algorithms on three test problems. For each problem below, we will offer a brief description of the problem itself. We will also specify for each one the set of parameters we have used.

4.1 Boolean Regression

We first use a simple two-input and one-output Boolean search problem [11]. This is a very simple prototypical GP problem in which all the absolute optimal solution is reachable at every repetition. We are therefore not interested in comparing the actual maximum fitness of our Lévy-flight GP against the traditional one, which is $fitness = 0$ in all cases, but rather, we will measure the speed at which this optimal function is found in terms of number of generations.

Table 1. LGP parameter configurations for Boolean search

Target function	x == y
Number of input	2
Number of output	1
Number of registers	2+2
Fitness function	Hamming distance
Operation set	{AND, OR, NAND, NOR}
Individual length (N)	16/24/32
Population size ($\|P\|$)	1000
Tournament size	4
Number of runs	10,000

The LGP parameter settings are shown in Table 1. In this problem, the LGP must replicate the behavior of a target Boolean function EQUALS form the possible 16 binary combinational logic functions: TRUE, FALSE, X, Y, AND, NAND, X IMPLY Y, Y IMPLY X, X NIMPLY Y, Y NIMPLY X, NOT X, NOT Y, OR, NOR, EQUALS, XOR. The reason EQUALS was selected is because it is the least probable function of all. More specifically, in our case with 2 calculation registers,

2 input registers, and 4 Boolean operators, we have $(2^2 \times 2^4 \times 2^4 \times 2^4)^I = (2^7)^I$ possible programs representing the 16 binary logic function, EQUALS is the least common of all.

We run independent simulations for all 3 individual length $N = \{16, 24, 32\}$ using fixed mutation rate $m = c/N$, where the constant $c = \{1, 2, 3, 4\}$ and compare these commonly-used parameter values with Lévy-flight mutation, where the rate m is drawn in a power-law distribution with exponents of $\gamma = \{1.0, 1.5, 2.0, 2.5, 3.0\}$. As the Boolean regression problem is *easy*, each independent run will evolve the absolute optimum in a "reasonable" number of generations. We record the number of generation each repetition until it reaches the optimal fitness and report averages and confidence intervals in Fig. 1. For reasons of space and readability, we only report result for $N = \{16, 24\}$. Trends for $N = 32$ are however consistent.

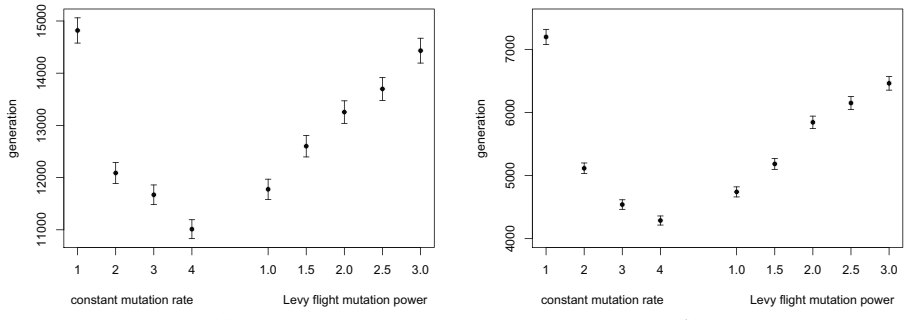


Fig. 1. Results of Boolean search problem. For each run, the total number of generations required to find the target is recorded. Results of conventional mutation scheme with a constant rate $m = c/N$ and Lévy-flight mutation with a power γ are compared for individuals of size (a) $N = 16$ and (b) $N = 24$. Points are mean values and error bars represent 95% confidence intervals.

In all three values of N, we observe a sharp fitness improvement with the increase of the constant c, thus with the increase of $m = c/N$. However, we see that mutation rate $m = 1/N$, which is the most commonly used in GP, is also the one that performs the poorest in the Boolean regression problem. The highest performance in this case is achieved with $c = 4$, for values of $c > 4$, the number of generations necessary for convergence increases (values not reported). With Lévy-flight mutation, the number of generations until convergence increases quasi linearly with increasing values of γ. This is explained by the fact that the power-law distribution becomes narrower (steeper on a log-log scale), which means that longer Lévy flights (i.e. higher mutation rate m) become less probable.

In order to assess the statistical significance of our results, we evaluate the results for each pair of c and γ values with a Kruskal-Wallis test and a a Bonferroni-Dunn Non Parametric for Multiple Comparison test [6,22]. The Kruskal-Wallis has always shown statistical differences between the groups ($p < 0.05$). The Bonferroni-Dunn test points out significant differences ($p < 0.05$) between mutation rate pairs. Results are reported

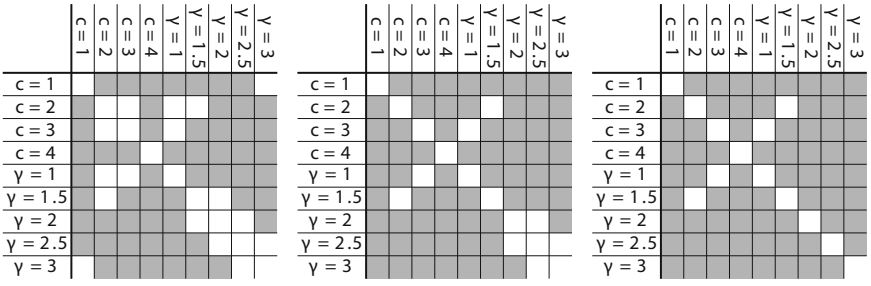


Fig. 2. Results of the statistical significance test for the boolean search problem. Matrix represent the results of the Bonferroni-Dunn Non Parametric for Multiple Comparison test for individuals of size (a) $N = 16$, (b) $N = 24$, and (b) $N = 32$. For each size, all possible combinations of c and γ are shown. Gray squares represent a statistically significant different in the statistical test. White squares represent non-significant differences.

in Fig. 2: gray squares represent a statistically significant different results, white squares represent non-significant differences.

For all individual sizes N, the Bonferroni-Dunn test highlight that most pairs of the mutation parameters c and γ shows statistically significant differences.

4.2 Numeric Regression: The Mexican Hat Function

For the first numeric search experiment, we choose a surface reconstruction and a polynomial regression [15]. This first problem consists in rebuilding the two-dimensional surface of the *mexican hat* function (Fig. 3) defined by the equation:

$$f_{mh}(x,y) = \left(1 - \frac{x^2}{4} - \frac{y^2}{4}\right) \times e^{\left(-\frac{x^2}{8} - \frac{y^2}{8}\right)}.$$

The parameters of this problem are specified in left column of Table 2. The fitness of each individual is evaluated as the sum of the squared errors between the predicted and the actual values of each point. This problem is much more difficult for LGP systems to solve. Because populations of LGP individuals do not reach optimal fitness, we report the average and confidence intervals for the fitness of the best individual in a population of 1000 individual after 2000 generations. We also perform the Bonferroni-Dunn statistical significance test on the results of each pairs of mutation rate c and γ. For reasons of space, we omit the figure reporting the fitness of individuals of size $N = 80$.

The average fitness improves as c (and $m = c/N$) increases until $c = 3$, for values of $c > 3$, the fitnesses deteriorates (not shown for readability reasons). As before, the worst results are obtained with a standard $m = 1/N$. In the case of Lévy flight mutations, fitness peaks $\gamma \approx 1.5 - 2.0$. Interestingly, $\gamma \approx 1$ yields worst fitness than higher values, which suggests that in this case, there are too many long Lévy flights for the system to converge to optima. Statistical significance test show that the results are different in many head-to-head c vs. γ comparison simulations.

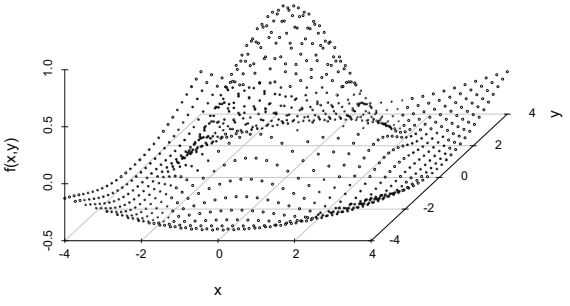


Fig. 3. The two-dimensional surface of the *mexican hat* function

Table 2. LGP parameter configurations for numeric search

Problem	mexican hat	quadratic polynomial
Number of input	2	1
Input range	$[-4.0,4.0]\times[-4.0,4.0]$	$[-1.0,1.0]$
Number of output	1	1
Sample size	400	100
Number of registers	$2+4$	$1+4$
Fitness function	sum of square errors	sum of square errors
Operation set	$\{+,-,\times,\div,x^y\}$	$\{+,-,\times,\div\}$
Constant set	$\{1,2,3,...,9\}$	$\{1,2,3,...,9\}$
Individual length (N)	40/60/80	40/60/80
Population size ($\|P\|$)	1000	1000
Tournament size	4	4
Number of runs	1000	1000

4.3 Numeric Regression: Quartic Polynomial Regression

The third and last problem we submit to our Lévy-flight mutation LGP system is a quartic polynomial regression of the form:

$$f(x) = x^4 + x^3 + x^2 + x$$

The LGP parameter settings for this problem are shown in the right column of Table 2. This is a problem of intermediate difficulty when compared to the 2 problem described above. Within the allocated maximal number of generations, a reasonable number of runs did in fact evolve the optimal solution. We report the results of numerical simulations in Figs. 5, 6, and 7 for individual sizes $N = \{40,60,80\}$ respectively. Left hand side panels, letter (a), show the cumulative distribution of the normalized number of simulation that successfully reach an optimal solution over the span of 5000 generations. For readability reasons, we limit the number of curves by reporting values $c = \{1,2,3\}$ and $\gamma = \{1.0,2.0,3.0\}$. Panels on the right-hand side show the results of the Bonferroni-Dunn test. Upper panels (b) show the statistically different results for simulation that

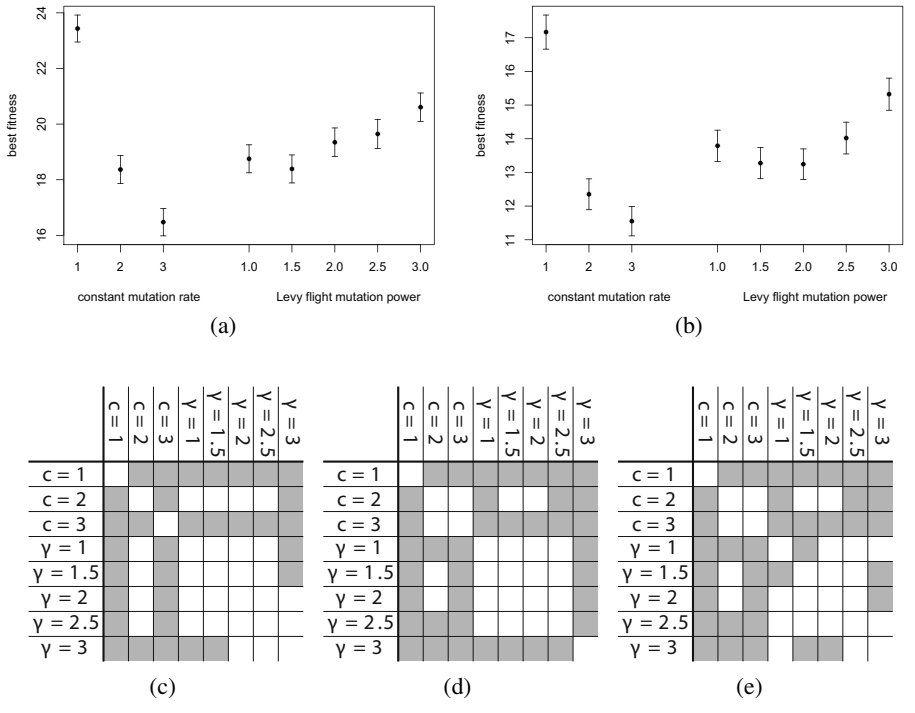


Fig. 4. Results of the mexican hat problem. Upper row, average best fitness after 2000 generations. Again, points are mean values and error bars represent 95% confidence intervals. Lower row, Bonferroni-Dunn test results comparing conventional mutation with a constant rate $m = c/N$ and Lévy-flight mutation with a power γ. Panels (a) and (c), individual $N = 40$. Panels (b) and (d), individual size $N = 60$. Panel (e), individuals size $N = 80$.

did evolve an optimal solution after the 5000 generation. Lower panels (c) are results of unsuccessful simulations, based on the best fitness evolved.

Though results are different, trends are similar across all values of N. Differences between curves only become magnified with the increase of N. Simulations with constant mutations rate $c = 1$ are persistently the worst performing, followed by the highest value of $\gamma = 3$. These are followed by the 2 intermediate values of $c = 2$ and $\gamma = 2.0$. The best results in this specific problem are obtained by a Lévy-flight mutation with $\gamma = 1.0$, closely followed by $c = 3$. In these cases, it is to be noted that although Lévy-flight mutations yield the best result at the end of the end of the evolutionary process, constant rate mutation show faster convergence (steeper slope). This remark doesn't hold as N increases. Statistical test show that most important number of pairs are, indeed, statistically different.

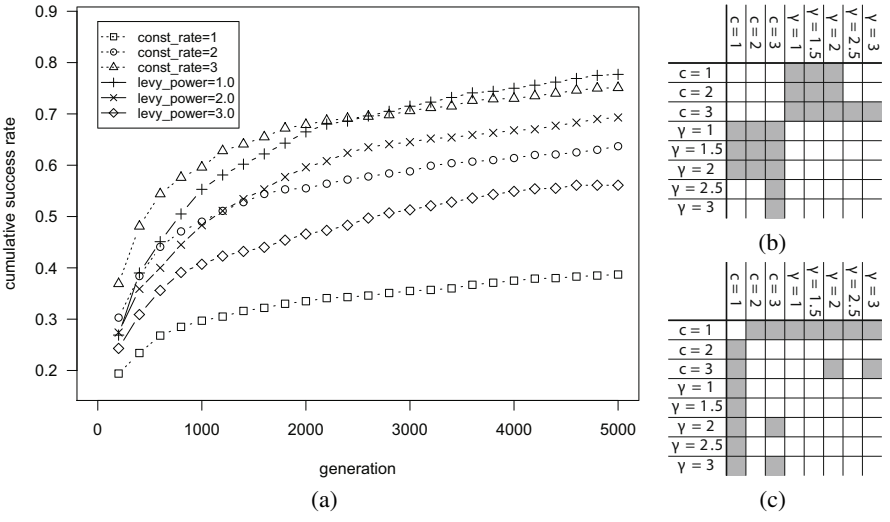


Fig. 5. Results of quartic polynomial regression problem for $N = 40$. (a) Cumulative success rate over 5000 generations. (b) statistical test results of successful simulations. (c) statistical test results of unsuccessful results.

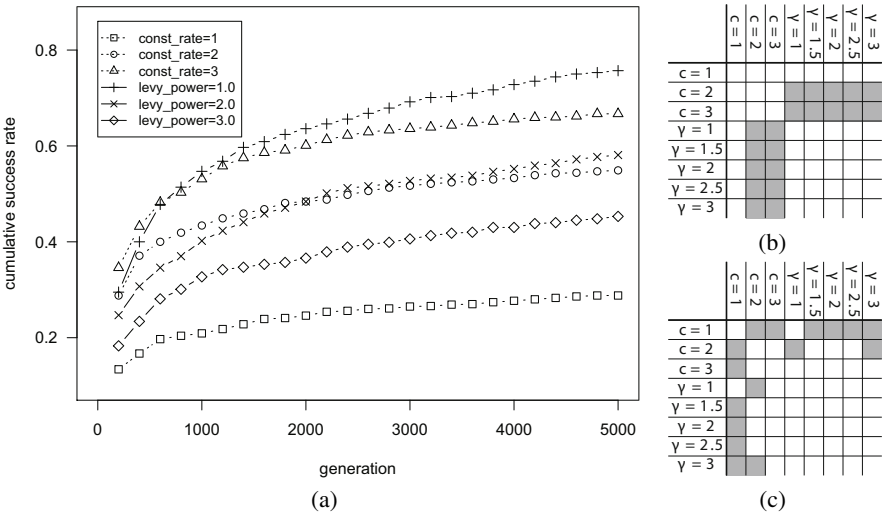


Fig. 6. Results of quartic polynomial regression problem for $N = 60$. (a) Cumulative success rate over 5000 generations. (b) statistical test results of successful simulations. (c) statistical test results of unsuccessful results.

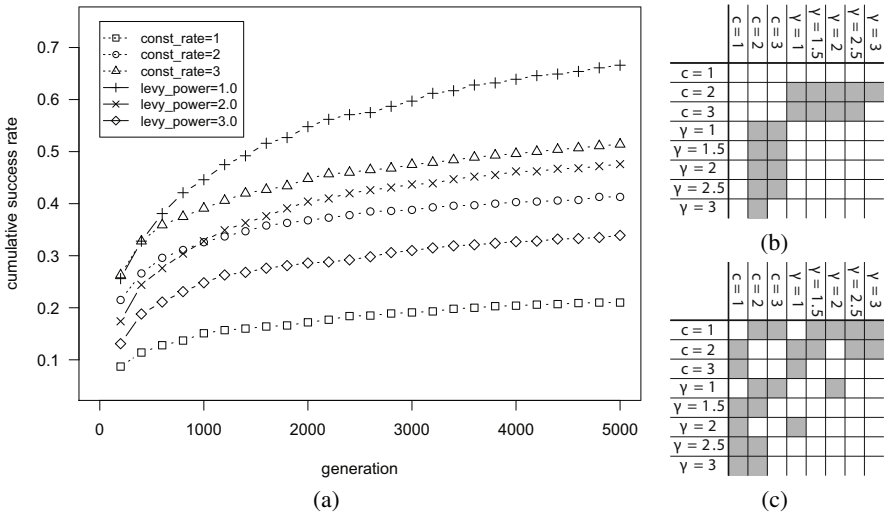


Fig. 7. Results of quartic polynomial regression problem for $N = 80$. (a) Cumulative success rate over 5000 generations. (b) statistical test results of successful simulations. (c) statistical test results of unsuccessful results.

5 Discussion, Conclusions, and Future Work

Although this is preliminary exploratory work, we demonstrate that in all problems, all values of γ in the Lévy flight mutation show higher performance than the $m = 1/N$ constant mutation rate that is usually used in genetic programing. We believe the behaviors observed in these experiments are interesting, and deserve further investigation in order to refine the model. Interestingly, the differences in performance for the different value of γ in Lévy-flight mutation are less pronounced, and it is therefore reasonable to assume we can reduce the parameter space of GP simulation by fixing γ to an intermediate value. In fact, we can easily tune the explorative behavior of our GP systems. The higher we fix the value of γ, the lower the probability of long Lévy flight. With lower values of γ, we increase the probability of long exploratory Lévy flights, without really reducing the short steps, that exploit the solution locally in the space.

We expect to generalize this work to other evolutionary algorithm, including genetic algorithm, on prototypical problems. We intend to study the effect of this mutation paradigm combined with recombination operators. In addition, we will apply the Lévy-flight mutation concept to a broader range of problem classes. Finally, we will be comparing our Lévy-flight variable mutation rate to other types of (self-)adaptive mutation rates that exist.

Acknowledgments. This work was partially supported by NIH grants LM009012, LM010098, AI59694, and by the Swiss National Science Foundation grant PBLAP3-136923. The authors are grateful to Luca Ferreri for his precious help withstatistical calculations and the corresponding figures, and to Joshua L. Payne for this invaluable contribution to the discussions.

References

1. Banzhaf, W., Nordin, P., Keller, R.E., Francone, F.D.: Genetic Programming – An Introduction: On the Automatic Evolution of Computer Programs and its Applications. Morgan Kaufmann, San Francisco (1998)
2. Benhamou, S., Bovet, P.: Distinguishing between elementary orientation mechanisms by means of path analysis. Animal Behaviour 43(3), 371–377 (1992)
3. Bovet, P., Benhamou, S.: Spatial analysis of animals' movements using a correlated random walk model. Journal of Theoretical Biology 131(4), 419–433 (1988)
4. Brameier, M., Banzhaf, W.: Linear Genetic Programming. Genetic and Evolutionary Computation, vol. XVI. Springer, Heidelberg (2007)
5. Cole, B.J.: Fractal time in animal behaviour: the movement activity of drosophila. Animal Behaviour 50(5), 1317–1324 (1995)
6. Dunn, O.J.: Multiple comparisons using rank sums. Technometrics 6(3), 241–252 (1964)
7. Edwards, A.M., Phillips, R.A., Watkins, N.W., Freeman, M.P., Murphy, E.J., Afanasyev, V., Buldyrev, S.V., Da Luz, M.G.E., Raposo, E.P., Stanley, H.E., et al.: Revisiting lévy flight search patterns of wandering albatrosses, bumblebeesand deer. Nature 449(7165), 1044–1048 (2007)
8. Edwards, A.M.: Overturning conclusions of lévy flight movement patterns by fishing boats and foraging animals. Ecology 92(6), 1247–1257 (2011)
9. Haupt, R.L., Haupt, S.E.: Practical Genetic Algorithms, pp, 2nd edn., pp. i–xvii. John Wiley & Sons, Inc. (2004)
10. James, A., Plank, M.J., Edwards, A.M.: Assessing lévy walks as models of animal foraging. Journal of the Royal Society Interface the Royal Society 8(62), 1233–1247 (2011)
11. Kantschik, W., Banzhaf, W.: Linear-Tree GP and Its Comparison with Other GP Structures. In: Miller, J., Tomassini, M., Lanzi, P.L., Ryan, C., Tetamanzi, A.G.B., Langdon, W.B. (eds.) EuroGP 2001. LNCS, vol. 2038, pp. 302–312. Springer, Heidelberg (2001)
12. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
13. Lee, C.Y., Yao, X.: Evolutionary programming using mutations based on the levy probability distribution. IEEE Transactions on Evolutionary Computation 8(1), 1–13 (2004)
14. Luke, S., Panait, L.: A comparison of bloat control methods for genetic programming. Evolutionary Computation 14(3), 309–334 (2006)
15. O'Neill, M., Vanneschi, L., Gustafson, S., Banzhaf, W.: Open issues in genetic programming. Genetic Programming and Evolvable Machines 11, 339–363 (2010), doi:10.1007/s10710-010-9113-2
16. Shlesinger, M., West, B., Klafter, J.: Lévy dynamics of enhanced diffusion: Application to turbulence. Physical Review Letters 58(11), 1100–1103 (1987)
17. Silva, S., Costa, E.: Dynamic limits for bloat control in genetic programming and a review of past and current bloat theories. Genetic Programming and Evolvable Machines 10(2), 141–179 (2009)
18. Vafaee, F., Nelson, P.: A genetic algorithm that incorporates an adaptive mutation based on an evolutionary model. In: International Conference on Machine Learning and Applications, ICMLA 2009, pp. 101–107 (December 2009)
19. Viswanathan, G.M., Afanasyev, V., Buldyrev, S.V., Stanley, H.E.: Lévy flights in random searches. Physica A 282, 1–12 (2000)
20. Viswanathan, G.M., Afanasyev, V., Buldyrev, S., Murphy, E., Prince, P., Stanley, H.E.: Lévy flight search patterns of wandering albatrosses. Nature 381(6581), 413–415 (1996)
21. Viswanathan, G.M., Buldyrev, S.V., Havlin, S., Da Luz, M.G., Raposo, E.P., Stanley, H.E.: Optimizing the success of random searches. Nature 401(6756), 911–914 (1999)
22. Zar, J.H.: Biostatistical Analysis, 5th edn. Pearson Prentice-Hall, Upper Saddle River (2010)

Understanding Zooplankton Long Term Variability through Genetic Programming

Simone Marini[1] and Alessandra Conversi[1,2]

[1] ISMAR - Marine Sciences Institute in La Spezia, CNR - National Research Council of Italy, Forte Santa Teresa, Loc. Pozzuolo, 19032 Lerici (SP), Italy
{simone.marini,a.conversi}@sp.ismar.cnr.it
http://www.ismar.cnr.it
[2] Marine Institute, University of Plymouth, Plymouth, PL4 8AA, United Kingdom

Abstract. Zooplankton are considered good indicators for understanding how oceans are affected by climate change. While climate influence on zooplankton abundance variability is currently accepted, its mechanisms are not understood, and prediction is not yet possible. This paper utilizes the Genetic Programming approach to identify which environmental variables, and at which extent, can be used to express zooplankton abundance dynamics. The zooplankton copepod long term (since 1988) time series from the L4 station in the Western English Channel, has been used as test case together with local environmental parameters and large scale climate indices. The performed simulations identify a set of relevant ecological drivers and highlight the non linear dynamics of the Copepod variability. These results indicate GP to be a promising approach for understanding the long term variability of marine populations.

Keywords: Ecological Modeling, Genetic Programming, Plankton Dynamics, Climate Change, Time Series.

1 Introduction

In a global warming scenario there is growing concern about the impact that climate change may have on marine ecosystems. In order to forecast such impacts it is crucial to understand, in particular, the effects of climate variability on zooplankton composition and productivity, as most marine animals belong to or feed on it during some stage of their life cycle.

Over the last two decades, several studies in different basins in the world have indicated that long-term (multidecadal) changes in zooplankton abundance and biomass are related to multidecadal changes in climate variability, but the causes behind this are still highly speculative [1].

An important step for understanding this relationship is the assembly of an appropriate computational model for explaining zooplankton variability. Most models used in biological oceanography are based on *a-priori* assumptions, and are usually based on correlation analysis, whether univariate or multivariate, and in few cases are based on linear or logistic regression. However, biological systems in

M. Giacobini, L. Vanneschi, and W.S. Bush (Eds.): EvoBIO 2012, LNCS 7246, pp. 50–61, 2012.

general, and zooplankton populations in particular, tend to not behave linearly, as the regime shifts recently observed in several marine systems [2].

A major difficulty in selecting ecological drivers is the identification of variables that are not individually relevant but that may become relevant in the context of others. On the contrary, features that are individually relevant may not all be useful because of possible redundancies [3]. Often, neither the experience of the researcher nor univariate or multivariate methods are sufficient to select relevant variables in a system. Another major problem in understanding ecological functioning is the choice of the statistical models. For instance, in [4] the zooplankton response to environmental changes is predicted through the logistic regression model, while in [5] an advection-diffusion-reaction equation is used to model the relationship between zooplankton and environmental variables. Of course, such *a-priori* assumptions reject other possible relationships among plankton and environmental drivers.

In this work the relation between plankton variability and environmental drivers is approached by choosing no *a-priori* drivers, and utilizing no *a-priori* functional forms. This goal is achieved by using the Genetic Programming (GP) approach [6,7]. The main characteristic that makes the GP approach suitable for this endeavour is the capability to generate functions able to approximate the investigated natural phenomenon without any strong *a-priori* assumption on the functional form or the (in)dependence of the involved variables. Moreover, the GP approach provides a natural selection of the relevant variables, and it is able to evolve mathematical models that are understandable, and whose interpretation can support the research activities of biologists and oceanographers. Even though the GP approach has been used in a growing range of applications dealing with marine environment [8,9,10], it has not yet been exploited for the description and prediction of plankton abundance variability, necessary steps for understanding the possible effects and scenarios due to climate change.

The marine area chosen for this study is the Western English Channel where, since 1988, zooplankton samples have been collected at the monitoring station L4, off Plymouth (UK) [11]. This ecosystem is characterized by both cold and warm temperate zooplankton species, and a large number of historical studies have been carried out [12]. The copepod time series measured here have been analyzed together with several ecological variables from the same site and climate indices affecting the northern hemisphere.

The paper is organized as follows. Section 2 discusses the data and methods used in this work, while section 3 shows the results obtained. Conclusions and future work on this study are presented in section 4.

2 Data and Methods

The aim of this research is to define a Symbolic Regression model, based on Genetic Programming (GP), able to express zooplankton abundance variability as function of climatological, physical and biological parameters. The functions generated by GP can be interpreted as approximations of the ecological mechanisms governing the target marine ecosystem.

2.1 Data

Table 1 shows the biotic and abiotic data used in this work.

Table 1. The environmental and climate variables used in this study

Time Series	Period	Data Gap	Source
Total Copepod	1988-2008	yes	WCO-L4[11]
Sea Surface Temperature (sst)	1988-2010	yes	
Salinity (sal)	1996-2010	yes	
Micro Zooplankton (mzp)	1992-2008	yes	
Chlorofill (chl)	1992-2011	yes	
Total Organic Carbon (toc)	1992-2011	yes	
Total Organic Nitrogen (ton)	1992-2011	yes	
North Atlantic Oscillation (NAO)	1950-2011	no	NOAA-CPC[13]
East Atlantic Pattern (EA)	1950-2011	no	
East Atlantic West Russia Pattern (EAWR)	1950-2011	no	
Scandinavian Pattern (SCA)	1950-2011	no	
Polar Eurasia Pattern (POL)	1950-2011	no	
North Hemisphere Temperature (NHT)	1850-2011	no	UEA-CRU[14]
Atlantic Multidecadal Oscillation (AMO)	1948-2011	no	NOAA-ESRL[15]

The focus of this study are marine copepods and in particular their variability over time. Copepods usually represent the largest mesozooplankton component and are found in all pelagic systems. They are an important food source for larval fish and for planktivorous pelagic fish, hence their variability has a direct impact on the food web. The total Copepod variable used in this work, see Figure 1, is the sum of the copepod abundances over all species identified in each sample and represent the copepod density in this area.

The sea surface temperature (sst) and the salinity (sal) are important physical parameters that are related to the sea current circulation. They do affect plankton biogeography, as plankton species have specific temperature and salinity ranges within which they can survive. Temperature has been shown to be of particular relevance in shaping plankton distributions and biogeographical shifts [1,12]. The chlorophyll-a (chl) is considered a proxy for phytoplankton biomass, which is an important food source for copepods. Microzooplankton (mzp) represents the smaller, unicellular, zooplankton fraction, and is also an important food source for copepods. Total organic carbon and nitrogen (toc and ton respectively) are proxies for primary production and for relevant nutrients for phytoplankton and their abundance may depend on anthropic activities. Details on the acquisition and measurements of these parameters can be found at [11].

Even if the investigated marine area is restricted to the L4 station, large scale climate patterns should need also to be considered for their influence on local scale. The Northern Atlantic Oscillation (NAO) is defined as the dipole of atmospheric pressure anomalies between Island (low) and Azores (high). Strong positive phases tend to be associated with above-average temperatures and precipitation over northern Europe in winter and below-average temperatures and

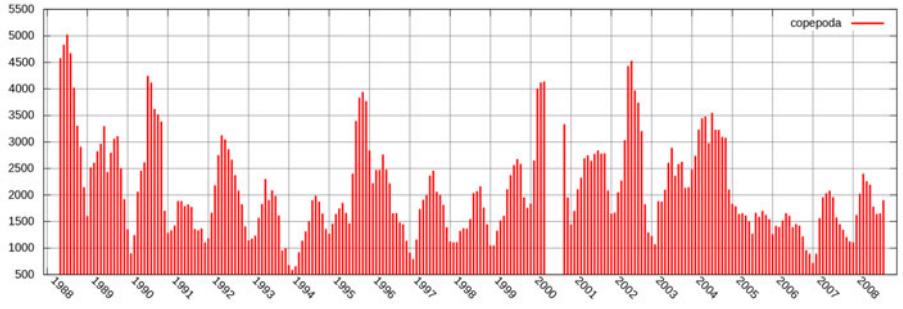


Fig. 1. Total Copepod abundance, bi-weekly sampling, 1988-2009; x-axis refers to years and y-axis refers to the number of individuals per m^3

precipitation across southern Europe. Opposite patterns of temperature and precipitation are observed during strong negative phases. The East Atlantic Pattern (EA) is structurally similar to the NAO but the anomaly centers are displaced southeastward with respect to the NAO. The positive phase of the EA is associated with above-average surface temperatures in Europe, with above-average precipitation over northern Europe and with below-average precipitation across southern Europe. The East Atlantic/West Russia Pattern (EAWR) affects the Eurasia continent where positive phases reflect below-average precipitation across central Europe. The Scandinavian Pattern (SCA) is also considered as one of the prominent indices of Eurasia. The positive phase is associated with below-average temperatures over western Europe and above-average precipitation across central and southern Europe, and below-average precipitation across Scandinavia. The Northern Hemisphere Temperature (NHT) index is defined as the combination of land and sea surface temperature anomalies [14] over the northern hemisphere, while the Atlantic Multidecadal Oscillation (AMO) describes long duration changes in the sea surface temperature of the North Atlantic Ocean. The AMO affects air temperatures and rainfall over much of the Northern Hemisphere, in particular, North America and Europe.

The time series presented in Table 1 refer to observed data with heterogeneous dimensionality and magnitude. In order to allow the GP based procedure to evolve the set of approximating functions, the data have been normalized and smoothed. Let $X = (x_1, \ldots, x_N)$ be a time series with time-indices $1, \ldots, N$, the normalization $\overline{X} = (\overline{x_1}, \ldots, \overline{x_N})$ is obtained by dividing every element x_i by the element of X with maximum value: $\overline{x_i} = \frac{x_i}{\max(X)}$. The smoothing filter reduces the noise from the time series replacing each value by its moving average computed with a smoothing period of l neighbors: $x_i = \frac{1}{2l} \sum_{k=i-l}^{k=i+l} x_k$. In this study $l = 2$.

2.2 Genetic Programming

The variables described above, will be utilized in a Symbolic Regression model based on the GP approach. The GP procedure generates solutions starting from

an initial population of randomly generated functions, then it improves the solutions by miming the selection processes that occur naturally in biological systems through Selection and the Crossover and Mutation genetic operators. The biological and physical time series, and the large scale climate indices are used as training set for the GP model. The Pyevolve libraries have been used for implementation [16].

The zooplankton dynamics are not necessarily linear and for this reason three different sets of mathematical operators, able to capture three different degrees of non-linearity, have been considered for the experiments:

$$\mathcal{S}_1 = \{+, -, *, /^*\}$$
$$\mathcal{S}_2 = \{+, -, *, /^*, sqrt^*, log^*, sin, cos, tan, atan\}$$
$$\mathcal{S}_3 = \{+, -, *, /^*, sqrt^*, log^*, sin, cos, tan, atan, min, max, if - then - else\}$$

where $/^*, sqrt^*, log^*$ operators correspond to protected division, protected square root and protected logarithm respectively. The term protected indicates a restriction on the mathematical operator such that $/^*$ returns 0 when a division by 0 is attempted, $sqrt^*$ returns 0 when a square root of a negative number is attempted and log^* returns a small number (10^{-100} in the performed experiments) if a logarithm of a number ≤ 0 is attempted. $\mathcal{S}_2$ adds trigonometric, logarithmic and square root operators to the basic set of mathematical operations in $\mathcal{S}_1$. $\mathcal{S}_3$ improves the non linear expressive potential of the approximating function by adding min and max. These operators return the minimum and the maximum between two values a and b. $\mathcal{S}_3$ also adds the $if - then - else$ operators defined as: $if\ a\ >\ b\ then\ c\ else\ d$ or $if\ a\ <\ b\ then\ c\ else\ d$, whose semantics is if a is greater than b then returns c else returns d, or if a is less than b then returns c else returns d, respectively.

The fitness function that has been used for the evaluation of the individuals is the Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - x'_i)^2} \tag{1}$$

where N is the number of values in the observed data and x and x' are the observed and the approximating values respectively.

Table 2 shows the GP parameters used in this study. This parameters have been selected, after several experiments, to reduce the overfitting effects.

2.3 Training and Validation and Genotype Analysis

The data described in section 2.1 are used to train the GP model described in Section 2.2. In particular, the variables of each individual are instantiated with the values of the corresponding time series and the fitness value is returned. On the other hand, time series have missing data and different begin-end dates. To avoid that some variables could not be instantiated, the training set TS is computed by intersecting all the time series, that is, by restricting all the time

Table 2. The Genetic Programming parameters used in this work

variables	sst, sal, mzp, chl, toc, ton, nao, ea, eawr, sca, pol, nht, amo
constants	0.0, 0.25, 0.5, 0.75, 1.0
initial population	ramped half and half
individual max depth	4
population size	1000
max generations	100
selector method	roulette wheel
crossover rate	0.9
mutation rate	0.02
elitism	the best individual is copied to the next generation
scaled fitness	linear scaling
termination criterion	max generations $\vee$ raw fitness = 0.00

series to the same begin-end dates and to the time-indices whose value exists for all the time series.

The training phase generates a set of functions corresponding to the individuals of the population that meet the termination criterion. Each generated function is an approximation of the zooplankton time series. Three experiments have been performed by using the three set of mathematical operators $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$. To reduce the probability that the approximation of the zooplankton time series is obtained by chance, each experiment consisted of 100 runs of the training phase, and at each run the best approximating function (smaller fitness value) was selected. Moreover, in order to reduce the impact of the missing data on the generated functions, only 75% of the training set time-indices has been randomly sampled at each run.

Let $\Phi_{\mathcal{S}_i}$ be the set of real valued functions obtained by performing the experiments $\mathcal{S}_i$ and let $\mathcal{V}$ be the set of variables listed in Table 1. Each function $f_{\mathcal{S}_i} \in \Phi_{\mathcal{S}_i}$ can be expressed as $f_{\mathcal{S}_i} : v_1 \times \cdots \times v_l \rightarrow \Re$, where $v_i \in \mathcal{V}$ with $1 \leq i \leq l$ and $l \leq |\mathcal{V}|$ is the number of variables occurring in $f_{\mathcal{S}_i}$. According to this definition the zooplankton abundance at the time t can be expressed as $z(t) = f_{\mathcal{S}_i}(v_1(t), \ldots, v_l(t))$, where $v_i \in \mathcal{V}$ and the time-index t is the $t-$th element of the training set time series associated to v_i, $1 \leq i \leq l$.

The validation phase is aimed at identifying which set of mathematical operators is the most effective for approximating the zooplankton time series, while the genotype of the elements in $\Phi_{\mathcal{S}_i}$ is analyzed to identify the most relevant variables and the most persistent functional forms responsible for the zooplankton dynamics.

Each function $f_{\mathcal{S}_i} \in \Phi_{\mathcal{S}_i}$ is validated singularly and the validation of the experiment $\mathcal{S}_i$ is obtained by averaging the validations of all the functions in $\Phi_{\mathcal{S}_i}$. The validation set VS, for $f_{\mathcal{S}_i}$, is obtained by intersecting all the time series associated to the variables $v_i \in \mathcal{V}$ occurring in $f_{\mathcal{S}_i}$. Since the training set is obtained by intersecting all the variables in $\mathcal{V}$ the set of time-indices of TS is contained into the set of VS time-indices, thus the validation phase can be used to estimate the predicting and generalization power of $f_{\mathcal{S}_i}$.

The genotype of the individuals belonging to $\Phi_{\mathcal{S}_i}$ is analyzed in order to identify which variables are most relevant in the experiment $\mathcal{S}_i$. These variables should be considered the ecological driver of the analyzed zooplankton community. In the proposed approach, the relevancy of each variable $v \in \mathcal{V}$ is defined as the number of functions that involves that variable with respect to the total number of functions. Finally the functional forms that involve the most relevant variables are analyzed in order to identify to which extent these variables contribute to increase or reduce the population of zooplankton in the investigated marine area.

3 Relevant Environmental Variables and Zooplankton Abundance Prediction

As discussed in section 2.3, three experiments have been defined based on the three set of mathematical operators $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$. The goals of the experiments are the identification of the most relevant environmental variables, the identification of the most effective mathematical operators needed for describing the zooplankton dynamics and the validation of the predicting power of the learnt approximating functions.

Three indicators have been defined to evaluate the approximating power of the learnt functions, as proposed in [17]. The Root Mean Squared Error (RMSE) defined in equation (1) indicates the average error of the approximation. The accuracy of the learnt function to follow the trend of the observed data is indicated by MISS, defined in equation (2):

$$MISS = \frac{1}{N-1} \sum_{i=2}^{N} \begin{cases} 1 \; if & \Delta x_i > 0 \wedge \Delta x'_i < 0 \\ & \vee \; \Delta x_i < 0 \wedge \Delta x'_i > 0 \\ & \vee \; \Delta x_i = 0 \wedge \Delta x'_i \neq 0 \\ & \vee \; \Delta x_i \neq 0 \wedge \Delta x'_i = 0 \\ 0 \; otherwise & \end{cases} \tag{2}$$

where $N = |\Phi_{\mathcal{S}_i}|$ is the number of functions in $\Phi_{\mathcal{S}_i}$, x and x' are the observed and the approximated values respectively and $\Delta x_i = x_i - x_{i-1}$. The Average Percentage Change (APC) indicating the magnitude of the incorrect prediction with respect to the trend of the observed data is defined in equation (3):

$$APC = \frac{1}{N-1} \sum_{i=2}^{N} \begin{cases} \frac{|x_i - x_{i-1}|}{x_{i-1}} \; if & \Delta x_i > 0 \wedge \Delta x'_i < 0 \\ & \vee \; \Delta x_i < 0 \wedge \Delta x'_i > 0 \\ & \vee \; \Delta x_i = 0 \wedge \Delta x'_i \neq 0 \\ & \vee \; \Delta x_i \neq 0 \wedge \Delta x'_i = 0 \\ 0 \qquad otherwise & \end{cases} \tag{3}$$

where Δx is defined as for equation (2). The smaller the values of RMSE, MISS and APC, the better is the approximation of the observed data.

Table 3 summarizes the approximation performance with respect to the three set of mathematical operators $\mathcal{S}_i$ and with respect to the training and validation

Table 3. Summary of the generalization performance

	training				validation			
	t-indices	**RMSE**	**MISS**	**APC**	**t-indices**	**RMSE**	**MISS**	**APC**
$\mathcal{S}_1$	44	0.103	0.259	0.070	109.19	0.170	0.358	0.050
$\mathcal{S}_2$	44	0.098	0.273	0.058	116.98	0.166	0.336	0.045
$\mathcal{S}_3$	44	0.107	0.236	0.068	111.98	0.153	0.324	0.038

phases. The columns t-indices represent the average number of time-indices of the training and validation set, as described in secton 2.3, while the values reported in the columns RMSE, MISS and APC are obtained by averaging the results of the equations (1), (2) and (3) over all the functions $f_{\mathcal{S}_i} \in \Phi_{\mathcal{S}_i}$, with $1 \leq i \leq 3$. The best generalization performance and predicting accuracy is provided by $\mathcal{S}_3$ and it is shown by the smallest values for RMSE, MISS and APC obtained during the validation phase. It is also noteworthy that the number of time-indices used to validate the functions are more than twice than the number of time-indices used in the training phase.

To understand the mechanism governing the zooplankton variability it is noteworthy to analyze the genotype of the individuals that belong to all three sets of functions $\Phi_{\mathcal{S}_i}$. Figure 2 shows the relevance score of the variables described in section 2.1 with respect to the three sets $\mathcal{S}_i$. The relevance of the variable v is obtained as the ratio between the number of individuals $f_{\mathcal{S}_i} \in \Phi_{\mathcal{S}_i}$ that involve v and the size of the population $|\Phi_{\mathcal{S}_i}|$ (in this study $|\Phi_{\mathcal{S}_i}| = 100$). The variables whose relevance fall between the two red lines have a probability to be generated by chance 5%. Such probability has been computed according to the binomial distribution $\binom{N}{k} p^k (1-p)^{N-k}$, where N is the number of runs ($N = 100$ in this study), k is the number of functions $f_{\mathcal{S}_i}$ in which the variable v occurs, and p is the probability that v happens within a function. In this case p has been considered uniform among the thirteen variables, that is $p = \frac{1}{13}$.

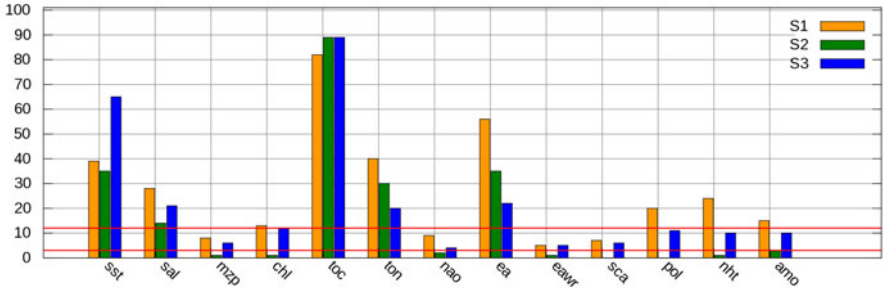


Fig. 2. Relevance of environmental variables discussed in section 2.1 with respect to the set of mathematical operators $\mathcal{S}_1, \mathcal{S}_2$ and $\mathcal{S}_3$. The y-axis represents the relavance score.

Five variables, out of thirteen, should be considered relevant with respect to the three sets of mathematical operators $\mathcal{S}_i$, namely the total organic carbon (toc), the sea surface temperature (sst), the east Atlantic pattern (ea), the total organic nitrogen (ton) and the salinity (sal). As shown in Figure 2, these variables obtain larger relevance score for all three experiments involving the set of mathematical operators $\mathcal{S}_i$. The other variables rarely occur in the learnt functions (low relevance score), have a non uniform behavior with respect to the three sets $\mathcal{S}_i$ and often have a probability to be selected by chance 5%. In literature it is known that zooplankton dynamics depends mainly on sst [12] while the influence of ea, toc, ton and sal has not yet been observed.

As shown in Table 4, the functions that involve only the five relevant variables (and do not involve at all the eight irrelevant variables) are 44% in $\Phi_{\mathcal{S}_1}$, 91% in $\Phi_{\mathcal{S}_2}$ and 64% in $\Phi_{\mathcal{S}_3}$. On the contrary, the functions that involve only irrelevant variables are 0% in all the three set of functions $\Phi_{\mathcal{S}_i}$. In $\Phi_{\mathcal{S}_2}$, 94% of the individuals involve only the operators $\sin, \cos, \tan, \arctan, \sqrt{\ }$ and $\log$, while the same set of operators is involved in 47% of $\Phi_{\mathcal{S}_3}$. The percentage of individuals of $\Phi_{\mathcal{S}_3}$ that involve at least one among $\min, \max$ and $if-then-else$ is 52%, while the same operators occur exclusively in only 6% of the individuals. These results

Table 4. Summary of the genotype analysis of the three set of individuals $\Phi_{\mathcal{S}_i}$

Genetic material	$\Phi_{\mathcal{S}_1}$	$\Phi_{\mathcal{S}_2}$	$\Phi_{\mathcal{S}_3}$
only toc, sst, ea, ton, sal	44%	91%	64%
only $mzp, chl, nao, eawr, sca, pol, nht, amo$	0%	0%	0%
only $\sin, \cos, \tan, \arctan, \sqrt{\ }, \log$	-	94%	47%
also $\min, \max, if-then-else$	-	-	52%
only $\min, \max, if-then-else$	-	-	6%

characterize the genotype of the populations resulting from the GP approach presented in section 2.2. Although the approach proposed for the identification of the relevance may be affected by introns, these results confirm the relevance of the variables shown in Figure 2. The most persistent mathematical operators are those of $\mathcal{S}_2$ and $\mathcal{S}_3$ as confirmed by $\Phi_{\mathcal{S}_2}$ and $\Phi_{\mathcal{S}_3}$ genotypes.

Table 3 indicates that the individuals of $\Phi_{\mathcal{S}_3}$ generalizes better than the individuals of $\Phi_{\mathcal{S}_2}$ and $\Phi_{\mathcal{S}_1}$, while the genotype analysis of all individuals, shown in Table 4, suggest that the generalization performance of $f \in \Phi_{\mathcal{S}_3}$ is obtained by introducing strong non linear operators like $\min, \max$ and $if-then-else$. The persistence of the trigonometric operators suggests the cyclic dynamics of the zooplankton community with respect to the environmental variables investigated. From the genotype analysis of $\Phi_{\mathcal{S}_2}$ it has been observed that zooplankton abundance increase as toc, sst, ton and sal increase. Moreover, 31 individuals out of 34, in $\Phi_{\mathcal{S}_2}$, involve the variable ea as part of the denominator or as argument of the cos function. This means that zooplankton abundance increases as ea tends to 0 and decrease otherwise. Similar behavior has been observed also for the individuals of $\Phi_{\mathcal{S}_3}$ that do not involve the $if-than-else$ operator. The equations

(4) and (5) are two examples from $\Phi_{\mathcal{S}_2}$ and $\Phi_{\mathcal{S}_3}$ that involve only relevant variables. In both the equations the zooplankton abundance z is proportional to toc and sst. Moreover, in eq. (4), large values of $|ea|$ make z decrease, and in eq. (5), z increase slower for large values of ea ($toc < ea$), indeed $cos(cos(toc)) \geq 0.54$, where $toc \in [0, 1]$ and $ea \in [-1, 1]$.

$$z(toc, sst, ea) = \cos(ea) * sst * \sqrt{\arctan(toc)} \tag{4}$$

$$z(toc, sst, ea) = \begin{cases} 0.54 * \max(\sin(sst), toc) & if\ toc < ea \\ cos(cos(toc)) * \max(\sin(sst), toc) & otherwise \end{cases} \tag{5}$$

The predicting power of the two individuals is summarized in Table 5.

Table 5. Summary of the approximating power of the individuals represented by equations (4) and (5)

	training				validation			
	t-indices	RMSE	MISS	APC	t-indices	RMSE	MISS	APC
eq. (4)	44	0.083	0.209	0.016	118	0.145	0.282	0.025
eq. (5)	44	0.098	0.116	0.026	118	0.140	0.299	0.027

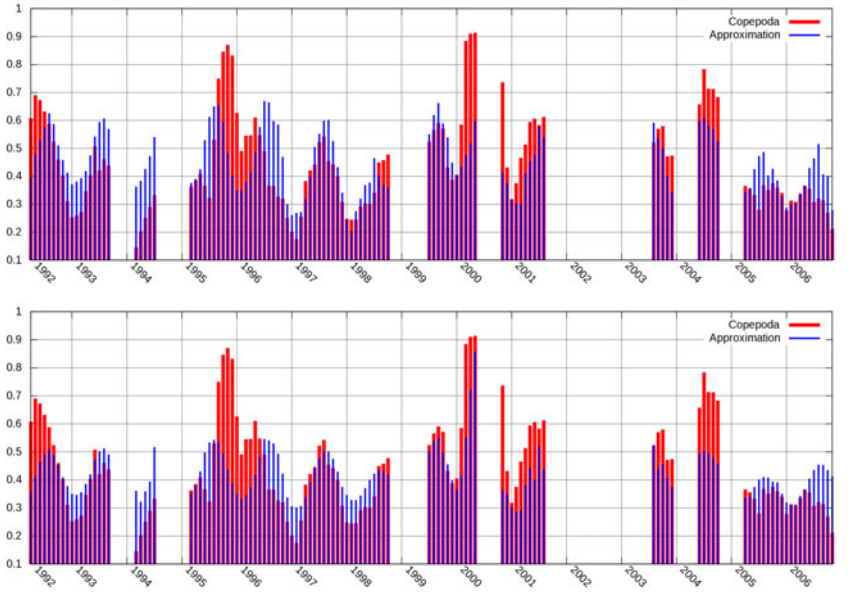


Fig. 3. Approximation of the copepods time-series; a) using the equation (4), b) using the equation (5)

Figure 3 shows the diagrams corresponding to the two individuals represented in the equations (4) and (5). The two individuals have been generated with a

training set whose time series have 44 time-indices, and validated with a validation set consisting of 118 time-indices. These two individuals exemplify the prediction accuracy provided by the relevant variables shown in Figure 2 and by the mathematical operators of $\mathcal{S}_2$ and $\mathcal{S}_3$. Most of the individuals $f \in \Phi_{\mathcal{S}_i}$ are not able to provide a good approximation of the zooplankton abundance in the period around 1996. This lack of approximation may depend on the gaps in the observed time-series and/or on some environmental variable not considered in this study.

4 Conclusion and Future Work

This work has utilized the copepods time series from the station L4 in the Western English Channel, to investigate decadal zooplankton variability with a novel methodology. The Genetic Programming approach has been used to express the copepods variability as function of several environmental variables, involving both large scale climate indices, and local physical and biological parameters. From the genotype analysis of the evolved functions emerges that total organic carbon (toc), sea surface temperature (sst), East Atlantic pattern (EA), salinity (sal) and total organic nitrogen (ton) are the most relevant parameters that drive the copepods variability in this area. In particular it has been noted that the copepod variability is subject to strong non linear dynamics that increases when toc, sst, sal and ton increase and |ea| tends to 0, and decreases otherwise. The dependence of the zooplankton dynamics from the previous environmental variables has not been observed in the literature and sheds an important and original insight in the mechanisms governing the zooplankton variability. As future work, the dynamics of other species of zooplankton from the same site, will be investigated and compared to the total copepod dynamics. Moreover the approximating capabilities of the Genetic Programming will be compared with other approaches based on multivariate regression, as for example the multivariate least square fitting and the Support Vector Regression (SVR) machine.

Acknowledgments. The local data used for this work come from the L4 station of the Western Channel Observatory [11] which is funded under the UK NERC National Capability. We are particularly grateful to Tim Smyth and to Claudia Halsband-Lenk for the information they provided on this data. We are thankful to NOAA-CPC, NOA-ESRL and UEA-CRU [13,15,14] for providing the climate data.

References

1. Beaugrand, G.: Decadal changes in climate and ecosystems in the north atlantic ocean and adjacent seas. Deep-Sea Research 56(8-10), 656–673 (2009)
2. Conversi, A., Umani, S.F., Peluso, T., Molinero, J.C., Santojanni, A., Edwards, M.: The mediterranean sea regime shift at the end of the 1980s, and intriguing parallelisms with other european basins. PLOS ONE 5(5) (2010)

3. Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L. (eds.): Feature Extraction, Foundations and Applications. Studies in Fuzziness and Soft Computing, vol. 207. Springer, Heidelberg (2006)
4. Marques, S., Azeiteiro, U., Leandro, S., Queiroga, H., Primo, A., Martinho, F., Viegas, I., Pardal, M.: Predicting zooplankton response to environmental changes in a temperate estuarine ecosystem. Marine Biology 155, 531–541 (2008)
5. Record, N., Pershing, A., Runge, J., Mayo, C., Monger, B., Chen, C.: Improving ecological forecasts of copepod community dynamics using genetic algorithms. Journal of Marine Systems 82(3), 96–110 (2010)
6. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
7. Poli, R., Langdon, W.B., McPhee, N.F.: A field guide to genetic programming (2008), http://www.gp-field-guide.org.uk
8. Muttil, N., Chau, K.W.: Machine-learning paradigms for selecting ecologically significant input variables. Eng. Appl. Artif. Intell. 20, 735–744 (2007)
9. Tung, C.P., Lee, T.Y., Yang, Y.C.E., Chen, Y.J.: Application of genetic programming to project climate change impacts on the population of formosan landlocked salmon. Environ. Model. Softw. 24, 1062–1072 (2009)
10. Ali Ghorbani, M., Khatibi, R., Aytek, A., Makarynskyy, O., Shiri, J.: Sea water level forecasting using genetic programming and comparing the performance with artificial neural networks. Comput. Geosci. 36, 620–627 (2010)
11. (WCO), http://www.westernchannelobservatory.org.uk/
12. Eloire, D., Somerfield, P.J., Conway, D.V.P., Halsband-Lenk, C., Harris, R., Bonnet, D.: Temporal variability and community composition of zooplankton at station l4 in the western channel: 20 years of sampling. Journal of Plankton Research 32(5), 657–679 (2010)
13. (NOA-CPC), ftp://ftp.cpc.ncep.noaa.gov/wd52dg/data/indices/tele_index.nh
14. (UEA-CRU), http://www.cru.uea.ac.uk/cru/data/temperature/
15. (NOA-ESRL), http://www.esrl.noaa.gov/psd/data/correlation/amon.us.long.-mean.data
16. Perone, C.S.: Pyevolve 0.6rc1, http://pyevolve.sourceforge.net/0_6rc1/
17. Iba, H., Nikolaev, N.: Genetic programming polynomial models of financial data series. In: Proc. of the Congress on Evolutionary Computation, pp. 1459–1466. IEEE Press (2000)

Inferring Disease-Related Metabolite Dependencies with a Bayesian Optimization Algorithm

Holger Franken[1], Alexander Seitz[1], Rainer Lehmann[2,3], Hans-Ulrich Häring[2,3], Norbert Stefan[2,3], and Andreas Zell[1]

[1] Center for Bioinformatics (ZBIT), University of Tübingen, D-72076 Tübingen, Germany
holger.franken@uni-tuebingen.de
[2] Division of Clinical Chemistry and Pathobiochemistry (Central Laboratory), University Hospital Tübingen, D-72076 Tübingen, Germany
[3] Paul-Langerhans-Institute Tübingen, Member of the German Centre for Diabetes Research (DZD), Eberhard Karls University Tübingen, Tübingen, Germany

Abstract. Understanding disease-related metabolite interactions is a key issue in computational biology. We apply a modified Bayesian Optimization Algorithm to targeted metabolomics data from plasma samples of insulin-sensitive and -resistant subjects both suffering from non-alcoholic fatty liver disease. In addition to improving the classification accuracy by selecting relevant features, we extract the information that led to their selection and reconstruct networks from detected feature dependencies. We compare the influence of a variety of classifiers and different scoring metrics and examine whether the reconstructed networks represent physiological metabolite interconnections. We find that the presented method is capable of significantly improving the classification accuracy of otherwise hardly classifiable metabolomics data and that the detected metabolite dependencies can be mapped to physiological pathways, which in turn were affirmed by literature from the domain.

1 Introduction

1.1 Background

Many diseases are the result of complex sequences of biochemical reactions involving several different metabolites. A central issue in the computational research field is understanding the biochemical interaction between various metabolites and the reconstruction of metabolic dependencies from metabolite patterns [16]. Non-alcoholic fatty liver disease (NAFLD) is a metabolic disease, which is known to be associated with insulin resistance, but can also be detected in insulin-sensitive subjects. Insulin-resistant individuals with NAFLD have a very high risk of developing type 2 diabetes (T2D) at an early stage whereas insulin-sensitive people with NAFLD are less likely to develop T2D [22,26]. In this work, we apply a method to select feature subsets that are relevant for the discrimination of samples from insulin-sensitive and -resistant individuals with NAFLD.

M. Giacobini, L. Vanneschi, and W.S. Bush (Eds.): EvoBIO 2012, LNCS 7246, pp. 62–73, 2012.

In parallel, we extract the information utilized by the method in order to gain insight into feature dependencies, which in this case correspond to interconnections between metabolites. In this way we try to improve the understanding of metabolic alterations that contribute to the development of T2D.

The young discipline of metabolomics has received increased attention in recent years. It measures small molecules contained in cells, tissues, or fluids involved in metabolism to reveal information about physiological processes. These measurements give a useful metabolic signature that represents normal biological processes, pathogenic processes, or responses to a therapeutic intervention [1].

Modern high-throughput techniques produce datasets that stand up to statistical scrutiny. This provides the opportunity to mine the generated data using statistical methods. The extraction of biologically relevant information and the reduction of factors such as noise, redundancy and dimensionality are central objectives of bioinformatics techniques. It has been shown that the classification accuracy of machine learning algorithms is not monotonic with respect to the addition of features, which in a targeted metabolomics approach, reflect calculated concentrations of predefined metabolites. Irrelevant or redundant features may degrade the predictive accuracy of the classification model. Feature selection is therefore focused on identifying and removing as much irrelevant or redundant information as possible [18].

Finding possible feature sets is a combinatorial optimization problem. This problem is generally NP-hard, as the space of feature subsets grows exponentially with the number of features. Techniques for feature selection can be organized into filter and wrapper methods. Filter methods, in most cases, compute a feature relevance score and discard low scoring features. The remaining subset of features serves as input to a classification algorithm. Filter techniques easily scale to very high-dimensional datasets, are computationally fast and are independent of the classification algorithm. A common disadvantage of filter methods is that most proposed techniques are univariate; i.e., each feature is considered separately, ignoring feature dependencies [24].

Wrapper methods search the feature subset space for the best subset. An individual in the search space is represented by a bitstring, each bit indicating whether a feature is present or absent. A classification algorithm is trained and the classification performance serves as an evaluation criterion for candidate feature subsets. In this manner, the search algorithm wraps around the classification model. Advantages of wrapper approaches include the interaction between feature subset search and model selection and the ability to take feature dependencies into account.

As randomized, evolutionary and population-based search algorithms, Genetic Algorithms (GAs) have been shown to be a good choice for finding small feature subsets with high discriminatory power [27]. GAs implicitly manipulate partial solutions of a problem, so called building blocks, by mechanisms of selection and recombination. They reproduce and mix building blocks without using information about dependencies among the related features. Recombination operators often break partial solutions, which can sometimes lead to losing them [12].

Estimation of Distribution Algorithms (EDAs) are a type of evolutionary algorithms that replace typical genetic operators with building a probabilistic model from promising solutions which encodes the dependencies among the variables of the problem. This model is an important source of information that can be exploited to assist in a better understanding of the underlying structure of the problem. The Bayesian Optimization Algorithm (BOA) uses a Bayesian Network (BN) to estimate the joint distribution of promising solutions and then samples new individuals from the BN [21]. Each node in the BN represents a feature and has two possible states (indicating absence or presence of the feature). The evolution of solutions is guaranteed by the factorization of the probability distribution of best individuals in each generation of the search. For this work, we apply a modified BOA. In addition to the selected features, we extract the probabilistic information that led to their selection.

1.2 State of the Art

As described in [24], wrapper methods using population-based search strategies have successfully been applied to a variety of tasks in bioinformatics. When the size of the problem allows for the application of the wrapper approach, several works have noted its superiority in terms of predictive accuracy [6]. In [12], an algorithm for feature subset selection by BN-based optimization was demonstrated to filter irrelevant and redundant features from artificial data sets and to achieve dimensionality reduction and improvement of classification accuracy in real data sets. Earlier studies demonstrated that wrapper approaches are capable of selecting sets of highly discriminative metabolite features as biomarker candidates [8].

For the automated inference of metabolite dependencies, some traditional approaches such as measuring correlation coefficients have been used, but nonlinear dependencies or dependencies among multiple features constitute a challenging problem. Relief is a feature weight-based algorithm [14]. Given training datasets, it detects those features that are statistically relevant to the target concept. The central mechanism of the Correlation-Based Feature Selection algorithm [10] is a heuristic that evaluates the value of subset features using average feature-class correlations and average feature-feature inter-correlations. Nicholson et.al. [2] show that metabolic data can be used to derive probabilistic graphical models of metabolite dependencies, and in [16], Suhre et. al. demonstrate that densely connected subgraphs in Gaussian graphical models can be attributed to known reactions in the human fatty acid metabolism. They showed that metabolite profiles can capture metabolic pathways.

2 Methods

We implemented BOA as described in [21] and made it available as part of the EvA2 framework [15]. EvA2 is a comprehensive metaheuristic optimization

framework with emphasis on Evolutionary Algorithms (EA) written in Java™. To perform feature selection, we created a modular Java software environment which integrates EvA2 and classification algorithms provided in WEKA [9].

2.1 The Bayesian Optimization Algorithm

As described in [21], the first population of solutions in BOA is generated randomly with a uniform distribution over the space of all possible solutions. Each individual in the population is then evaluated by a classification algorithm and a set of promising solutions is selected. Following the suggestion in [21], we apply a range-based approach selecting the best $N/2$ from the N individuals of the population. If the number of selected solutions is close to N, the populations will not evolve very much from generation to generation. On the other hand, a low number will lead to low diversity. The selected set is used for building a BN. After the network is built, it is used to generate new candidate solutions. Finally, the new solutions are incorporated into the population after removing the least promising ones. We halt the algorithm after the completion of 30,000 executions of the evaluation function.

Bayesian networks (BNs). BNs are graphical models that can represent conditional probabilities among multiple variables of a problem. A BN has two components: structure and parameters. The structure is a directed acyclic graph (DAG), in which each node represents a variable of the underlying problem and the edges between nodes represent probabilistic dependencies among the variables. Parameters are the conditional probabilities computed for each of the variables according to the different value combinations of their parents. A BN can be used to generate new instances with similar properties as those of given data. [3].

Network construction. Finding the best network for a problem requires estimating the graph topology and its parameters. The goal is to learn a BN that best explains the given training data [3]. As it was shown in [4], this task is NP-complete. Learning the network structure requires a search procedure and scoring metric. The search procedure explores the space of all possible networks to find the one that maximizes the scoring metric. For reasons of computational feasibility, we reduce the space of networks by restricting the number of incoming edges into each node to three.

A greedy algorithm is used to find the best network. It starts with an empty network, and in each step, adds the edge that maximally increases the metric score while maintaining an acyclic topology. The algorithm stops when no further improvements in the network score are possible. Networks constructed in this way are not guaranteed to be the optimum representation of the underlying data, but encode the probability distribution of promising solutions to a reasonable extent. Methods that construct the exact optimum network for a problem entail huge computational requirements and are therefore only suitable for small problem sizes [7].

To discriminate between different networks, a scoring metric is needed to measure how well the BN models the data. From the class of Bayesian metrics, we select the frequently used Bayesian-Dirichlet metric (BDM) [20]:

$$p(D, B|\xi) = p(B|\xi) \prod_{i=1}^{n} \prod_{\pi x_i} \frac{\Gamma(m'(\pi_{X_i}))}{\Gamma(m'(\pi_{X_i}) + m(\pi_{X_i}))} \cdot \prod_{x_i} \frac{\Gamma(m'(x_i, \pi_{X_i}) + m(x_i, \pi_{X_i}))}{\Gamma(m'(x_i, \pi_{X_i}))}$$

Here

- Γ represents the Gamma function: for integers $\Gamma(x) = (x-1)!$
- $p(B|\xi)$ is the initial probability of the network B given optional prior information ξ
- $m(\pi_{X_i})$ is the number of instances in the data D where the parent nodes of the i^{th} bit are set to π_{X_i}
- $m'(\pi_{X_i})$ denotes optional prior information about the number of instances where the parent nodes of the i^{th} bit are set to π_{X_i}
- $m(x_i, \pi_{X_i})$ is the number of instances in D that have their i^{th} bit set to x_i and their parent nodes set to π_{X_i}
- $m'(x_i, \pi_{X_i})$ denotes optional prior information about the number of instances that have their i^{th} bit set to x_i and their parent nodes set to π_{X_i}.

When we assume no prior information about the number of instances with a special configuration for their variables, $m'(x_i, \pi_{X_i})$ can be set to 1 for all possible values. The resulting version of the BDM is known as K2, which is the second metric we consider in this work.

From the class of minimum description length metrics, we use the Bayesian Information Criterion (BIC):

$$BIC(B) = \sum_{i=1}^{n} \left(\sum_{\pi_{X_i}} \left(\sum_{x_i} (m(x_i, \pi_{X_i}) \log(m(x_i, \pi_{X_i}))) - m(\pi_{X_i}) \log(m(\pi_{X_i})) \right) - \frac{1}{2} \log(N) |x_i| \, |\pi_{X_i}| \right)$$

BIC tries to prevent the models from overfitting to the given data by using the penalization term $\frac{1}{2} \log(n) |x_i| \, |\pi_{X_i}|$. This term increases with the model complexity and therefore favors networks with fewer parameters.

Generating new candidate solutions. When the construction of the BN is complete, new individuals are generated from the network. First, for each variable, the conditional probabilities of each possible instance given all possible instances of its parents are computed. A well-defined ordering of the nodes into parents and children is guaranteed to exist due to the acyclic topology. Traversing through the network, each bit in a new instance is then set according to the conditional probability of the corresponding node. This procedure is repeated until $N/2$ new instances are created.

Modified BOA. To track the feature dependencies during the optimization process, we modified the BOA procedure. In our implementation, the BNs, which are derived from the most promising solutions in each generation, are extracted and stored. After termination of the algorithm, we compute the relative frequency for every edge that was ever established during the optimization procedure. The idea is that dependencies between strongly connected features are detected in many iterations of the algorithm. Hence, edges that connect such features are frequently established. A ranking of all edges is obtained by sorting them according to their relative frequency.

We carried out this ranking procedure for every combination of classifier and metric and performed five repetitions resulting in five edge-rankings for each combination. To achieve a measure of reproducibility, we computed Kendall's coefficient of concordance (Kendall's W) between these five rankings. Kendall's W is a descriptive statistic indicating the strength of the agreement among the the rankings. If W is 1, all the rankings are concordant; i.e., each edge received the same rank in each of the five repetitions. If W is 0, then there is no overall trend of agreement and the rankings may be regarded as random.

Classifiers. From WEKA, we applied different classifier types:

- k-nearest-neighbor (kNN): an instance-based learner, which compares each new instance to existing ones using a distance metric
- K*NN: another instance-based method with an entropic distance measure [5]
- Naive Bayes: a classifier-based on Bayes' formula for conditional probabilities
- J4.8 decision tree: a reimplementation of the C4.5 decision tree, which has been shown to have a very good combination of error rate and speed [17]
- Random Forest: a metalearner that bags ensembles of random trees
- Linear SVM: a maximum-margin-based classifier, which we chose due to its amenities concerning interpretability compared to nonlinear models [25]

We want our selected features to be interpretable by a domain expert. Hence, we focused on classification methods that allow conclusions to be drawn about the involved features. The modular structure of our software environment allows us to combine the modified BOA search procedure with an arbitrary classification method and thereby examine the compatibility of the different classifier types with BOA and their impact on the results.

For the evaluation of each feature subset, we perform a stratified nested cross validation. The optimal parameter combination is determined in an inner loop within a two-fold cross-validation. The performance of the selected parameters is then evaluated in an outer loop using three-fold cross-validation according to the area under the receiver operating characteristic curve (AUC). We carry out this validation scheme five times to avoid bias induced by the random number generator and then compute the average AUC.

2.2 Experimental Design

Plasma samples of 40 adults with NAFLD (20 insulin-sensitive and 20 -resistant subjects) were analyzed by a targeted metabolomics approach. All individuals

were intensively phenotyped as part of the Tübingen Lifestyle Intervention Program (TULIP) and considered healthy according to physical examination and routine laboratory tests.

Biocrates (Innsbruck, Austria) measured the concentrations of 247 compounds in EDTA-plasma by targeted IDQ. This targeted metabolomics analytical platform combines flow injection, liquid chromatography, and gas chromatography mass spectromectric approaches. We consider a measured concentration to be reliable if it exceeds a signal-to-noise ratio of three. To ensure validity we only consider metabolites that contained at least 70 % reliable measurements. This restriction excludes 69 metabolites from the data set, leaving 178 compounds for the data analyses (amino acids, acylcarnitines, bile acids, free fatty acids, sphingomyelins, phosphatidylcholines and lysophosphatidylcholines). For the computational analyses the data is mean centered and scaled to unit variance.

We apply each of the classifiers and metrics introduced in Sec. 2.1 in the modified BOA. In addition to improving the classification accuracy of insulin-sensitive and -resistant subjects with NAFLD we reconstruct networks from the detected feature dependencies and examine whether they represent physiological metabolite interconnections.

3 Results and Discussion

3.1 Classification Accuracy

The first question we addressed is whether BOA can enhance the classification performance when it is used as wrapper in a feature selection approach. Fig. 1 (a) displays the classification performances of the applied classifiers. It shows the distribution of average AUCs that all classifiers achieved when applied to the best feature set selected by BOA and the distribution of average AUCs all classifiers achieved when applied to the complete data set (NoFS).

Fig. 1 (a) demonstrates that classification performances without feature subset selection are close to an AUC of 0.5 and were significantly improved by BOA, indicating that it is capable of selecting discriminatory feature subsets. The most discriminative subset yields an average AUC of 0.93 and was selected using the K*NN classifier and the BDM for network construction. Further, the weakest discriminative power with an average AUC of 0.70 is revealed by a feature subset also selected using the BDM. The classifier, in this case, is the Random Forest. By further investigating the results from individual classifiers and metrics, we address the following questions: Do the applied classifiers differ in their performance and does the metric used in constructing the BN impact the achieved classification accuracy?

In Fig. 1 (b) we compare the different classifiers. Each box denotes the distribution of average AUCs across all results that were achieved using the respective classifier. K*NN and Naive Bayes yield the highest classification accuracies, followed by the k-nearest-neighbor learner and the linear SVM. The results obtained by the decision tree and the Random Forest are comparable to each other, but considerably worse than the other classifiers.

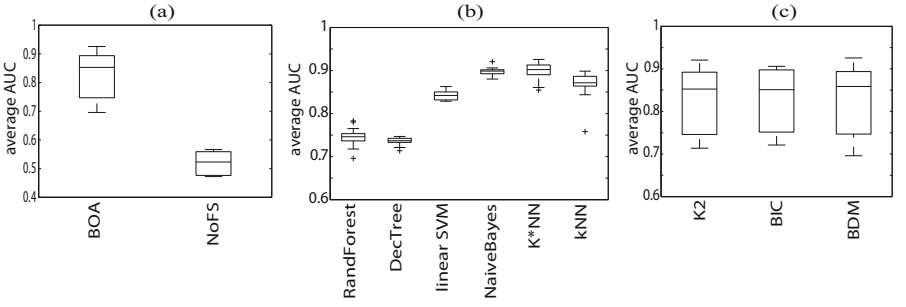


Fig. 1. Comparison of the classification performances (a) before and after feature subset selection; (b) of the applied classifiers; (c) using different scoring metrics

Fig. 1 (c) displays a comparison of the results that were obtained using the different scoring metrics. It shows the distributions of average AUCs all classifiers achieved when the respective metric was applied. No considerable difference can be observed between these results. For each metric, the distribution covers the complete range of classification accuracies that are obtained by the different classifiers. These results indicate that the choice of a scoring metric does not measurably influence the level of predictive accuracy that can be achieved by BOA-based feature subset selection.

3.2 Reproducibility of Edges

As described in section 2.1, for each combination of metric and classifier, we determined Kendall's W between rankings of edges based on their relative frequency during five repetitions of the optimization process. The results of that examination are presented in Fig. 2.

Each box in Fig. 2 (a) summarizes the distribution for a fixed classifier while varying the scoring metrics. The Random Forest classifier achieves the best results in this respect yielding a maximum W of 0.53 in combination with the BDM. With regard to the best achieved value, it is followed by K*NN and the Naive Bayes classifiers. However, it must be stated that the differences between the individual classifiers are not significant.

Similarly, each box in 2 (b) represents the distribution of W for a fixed metric while varying classifiers. The results show that the BDM performs considerably better than the K2 metric and both of them perform significantly better than the BIC. The multiruns in which BDM was used show a median W of 0.48, which indicates a reasonable degree of agreement.

These results demonstrate that the choice of the scoring metric has an impact on the reproducibility of the edges in the BNs that are generated during the optimization process. In contrast, the reproducibility is not strongly affected by the choice of the classifier.

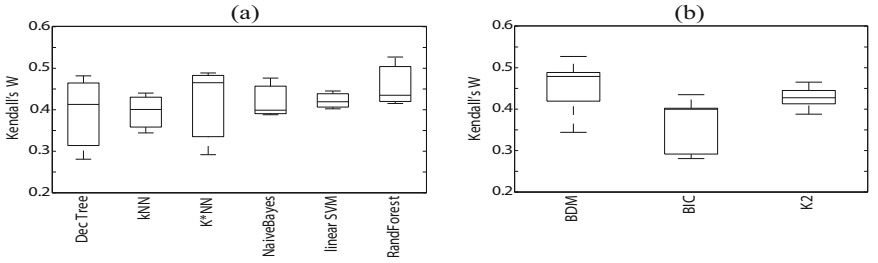


Fig. 2. Kendall's coefficient of concordance between the edge rankings using (a) the applied classifiers and (b) the different metrics

3.3 Networks of Dependent Metabolites

In the following, we reconstructed networks of metabolites that are connected by the most relevant edges arising from the above described results. Based on our findings in Sec. 3.1 and 3.2, we used edges that were generated using the K*NN and the Naive Bayes classifiers for the network reconstruction as these yielded the most accurate classifications. Due to its superiority regarding edge reproducibility, we chose the BDM as the scoring metric.

To extract the most relevant edges, we first discarded those that were established in less than 4 out of the 5 repetitions. For each of the remaining edges we computed the mean of its relative frequencies during the optimization process and created a ranking. For the network reconstruction we selected the top 5 % from that ranking.

In this way, we reconstructed two networks: one from the results using the K*NN classifier and one from the results using Naive Bayes. The one created using K*NN includes 78 metabolites in total, grouped by the edges to form 20 separate components of different sizes: 11 components of size 2, 5 components of size 3, 3 components of size 4 and one major component including 29 metabolites. The network constructed using Naive Bayes includes 79 metabolites, grouped to 19 components: 8 components of size 2, 2 components of size 3, 5 components of size 4, 1 component of size 5, 2 components of size 6 and again one major component of size 20.

3.4 Physiological Interpretation

To assess whether the reconstructed networks reflect physiological metabolite interconnections, we performed an automated lookup in the KEGG database [13]. For each connected component, we looked up the contained metabolites and the human metabolic pathways they are involved in. Then, for each pathway, we counted the number of metabolites it shares with the considered connected component as a test statistic. As a baseline, we then constructed 100 sets of metabolites of the same size as the considered network component, each of which was selected randomly from our data set. These yielded an empirical distribution of

Table 1. Significant overlaps between network components and metabolic pathways

Pathway	p-value in component			
	NB20	NB6a	NB6b	K*NN29
Linoleic acid metabolism	**0.01**	0.77	0.50	0.52
Valine, leucine and isoleucine biosynthesis	1	**0.03**	1	1
Sphingolipid metabolism	1	1	1	**0.03**
Glycerophospholipid metabolism	**0.04**	0.81	0.55	0.21
alpha-Linolenic acid metabolism	**0.04**	0.74	0.48	0.46
Arachidonic acid metabolism	**0.05**	0.74	0.48	0.47

test statistics for which the null hypothesis is true; i.e., the observed intersection between pathway and network component was achieved by random selection of metabolites. The intersection obtained from the reconstructed network component is then compared with this empirical distribution to assess significance. For the pathways that exceeded the significance threshold of 0.05, the p-values are given in Tab. 1. The term *NB20* in the table denotes the component of cardinality 20 in the network that was reconstructed using Naive Bayes. *NB6a* and *NB6b* encode components of size 6 within the same network. Analogously, *K*NN29* denotes the component of size 29 in the network established using the K*NN classifier. We only applied the significance testing to network components of a minimal cardinality of 6 as no meaningful significance can be expected for smaller numbers of metabolites.

Tab. 1 displays that three out of the four tested connected components show significant overlap with one or more pathways. Interestingly the detected pathways are referred to in the literature with regard to their role in T2D and NAFLD. The linoleic acid, glycerophospholipid, alpha-linolenic acid and arachidonic acid metabolism, which show significant overlap with the *NB20* component are discussed in [23]. The sphingolipid metabolism, which exhibits significant overlap with the *K*NN29*, component is also addressed by the authors. The component *NB6a* significantly overlaps with the valine, leucine and isoleucine biosynthesis. These branched-chain amino acids were also found by Newgard et. al. in [11] and [19] to be strongly associated with insulin sensitivity and may contribute to the development of insulin resistance and diabetes.

4 Conclusions

Our results demonstrate that feature subset selection using BOA is capable of significantly improving the classification accuracy of otherwise hardly classifiable metabolomics data. Using this approach, the choice of the applied classification algorithm strongly influences the results regarding classification performance. Additionally, the feature dependency information extracted by the modified BOA provides automated identification of metabolite interconnections. With respect to reproducibility, the choice of the scoring metric used in the network construction phase of BOA was shown to be important. The detected

metabolite dependencies can be mapped to physiological pathways which in turn were confirmed by literature from the domain of NAFLD and T2D research. The results suggest that the presented approach could also be used to generate hypotheses on metabolite interactions that are not yet known to play a role in disease processes.

Acknowledgements. This investigation was supported in parts by the Kompetenznetz Diabetes mellitus (Competence Network for Diabetes mellitus) funded by the Federal Ministry of Education and Research (FKZ 01GI0803-04) and a grant from the German Federal Ministry of Education and Research to the German Center for Diabetes Research (DZD eV).

References

1. Atkinson, A., Colburn, W., DeGruttola, V., DeMets, D., Downing, G., Hoth, D., Oates, J., Peck, C., Schooley, R., Spilker, B., et al.: Biomarkers and surrogate endpoints: Preferred definitions and conceptual framework. Clinical Pharmacology & Therapeutics 69(3), 89–95 (2001)
2. Bang, J., Crockford, D., Holmes, E., Pazos, F., Sternberg, M., Muggleton, S., Nicholson, J.: Integrative top-down system metabolic modeling in experimental disease states via data-driven Bayesian methods. The Journal of Proteome Research 7(2), 497–503 (2008)
3. Ben-Gal, I.: Bayesian networks. Encyclopedia of Statistics in Quality and Reliability (2007)
4. Chickering, D.: Learning Bayesian networks is NP-complete. Learning from data: Artificial intelligence and statistics 112, 121–130 (1996)
5. Cleary, J., Trigg, L.: K*: An Instance-based Learner Using an Entropic Distance Measure. In: Proceedings of the 12th International Conference on Machine Learning, pp. 108–114 (1995)
6. Doak, J.: An evaluation of feature-selection methods and their application to computer security (Technical Report CSE-92-18). Davis: University of California, Department of Computer Science (1992)
7. Echegoyen, C., Lozano, J., Santana, R., Larranaga, P.: Exact Bayesian network learning in estimation of distribution algorithms. In: IEEE Congress on Evolutionary Computation, CEC 2007, pp. 1051–1058. IEEE (2007)
8. Franken, H., Lehmann, R., Häring, H., Fritsche, A., Stefan, N., Zell, A.: Wrapper- and Ensemble-Based Feature Subset Selection Methods for Biomarker Discovery in Targeted Metabolomics. Pattern Recognition in Bioinformatics, 121–132 (2011)
9. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.: The WEKA data mining software: an update. ACM SIGKDD Explorations Newsletter 11(1), 10–18 (2009)
10. Hall, M.: Correlation-based Feature Selection for Machine Learning. Ph.D. thesis, The University of Waikato (1999)
11. Huffman, K., Shah, S., Stevens, R., Bain, J., Muehlbauer, M., Slentz, C., Tanner, C., Kuchibhatla, M., Houmard, J., Newgard, C., et al.: Relationships between circulating metabolic intermediates and insulin action in overweight to obese, inactive men and women. Diabetes Care 32(9), 1678 (2009)

12. Inza, I., Larranaga, P., Etxeberria, R., Sierra, B.: Feature subset selection by Bayesian network-based optimization. Artificial Intelligence 123(1-2), 157–184 (2000)
13. Kanehisa, M., Goto, S., Furumichi, M., Tanabe, M., Hirakawa, M.: Kegg for representation and analysis of molecular networks involving diseases and drugs. Nucleic Acids Research 38(Database issue), D355–D360 (2010)
14. Kira, K., Rendell, L.: The feature selection problem: traditional methods and a new algorithm. In: Proceedings of the Tenth National Conference on Artificial Intelligence, pp. 129–134. AAAI Press (1992)
15. Kronfeld, M., Planatscher, H., Zell, A.: The EvA2 optimization framework. Learning and Intelligent Optimization, 247–250 (2010)
16. Krumsiek, J., Suhre, K., Illig, T., Adamski, J., Theis, F.: Gaussian graphical modeling reconstructs pathway reactions from high-throughput metabolomics data. BMC Systems Biology 5, 21 (2011)
17. Lim, T.: A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-Three Old and New Classification Algorithms. Machine Learning 40, 203–228 (2000)
18. Masseglia, F., Poncelet, P., Teisseire, M.: Successes and new directions in data mining. Information Science Publishing (2008)
19. Newgard, C., An, J., Bain, J., Muehlbauer, M., Stevens, R., Lien, L., Haqq, A., Shah, S., Arlotto, M., Slentz, C., et al.: A branched-chain amino acid-related metabolic signature that differentiates obese and lean humans and contributes to insulin resistance. Cell Metabolism 9(4), 311–326 (2009)
20. Pelikan, M., Goldberg, D.: Hierarchical bayesian optimization algorithm, vol. 33, p. 63. Springer, Heidelberg (2006)
21. Pelikan, M., Goldberg, D., Cantu-Paz, E.: BOA: The Bayesian optimization algorithm (IlliGAL Report No. 99003). University of Illinois at Urbana-Champaign, Urbana (1999)
22. Petersen, K., Dufour, S., Befroy, D., Lehrke, M., Hendler, R., Shulman, G.: Reversal of Nonalcoholic Hepatic Steatosis, Hepatic Insulin Resistance, and Hyperglycemia by Moderate Weight Reduction in Patients With Type 2 Diabetes. Metabolism 54, 603–608 (2005)
23. Puri, P., Baillie, R.A., Wiest, M.M., Mirshahi, F., Choudhury, J., Cheung, O., Sargeant, C., Contos, M.J., Sanyal, A.J.: A lipidomic analysis of nonalcoholic fatty liver disease. Hepatology 46(4), 1081–1090 (2007)
24. Saeys, Y., Inza, I.N., Larrañaga, P.: A review of feature selection techniques in bioinformatics. Bioinformatics 23(19), 2507–2517 (2007)
25. Schölkopf, B., Smola, A.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. Adaptive Computation and Machine Learning, 1st edn. The MIT Press (2001)
26. Stefan, N., Kantartzis, K., Häring, H.U.: Causes and metabolic consequences of Fatty liver. Endocrine Reviews 29(7), 939–960 (2008)
27. Zou, W., Tolstikov, V.: Probing genetic algorithms for feature selection in comprehensive metabolic profiling approach. Rapid Communications in Mass Spectrometry 22(8), 1312–1324 (2008)

A GPU-Based Multi-swarm PSO Method for Parameter Estimation in Stochastic Biological Systems Exploiting Discrete-Time Target Series*

Marco S. Nobile[1], Daniela Besozzi[2], Paolo Cazzaniga[3], Giancarlo Mauri[1], and Dario Pescini[4]

[1] Università degli Studi di Milano-Bicocca, Dipartimento di Informatica, Sistemistica e Comunicazione, Viale Sarca 336, 20126 Milano, Italy
{nobile,mauri}@disco.unimib.it
[2] Università degli Studi di Milano, Dipartimento di Informatica e Comunicazione Via Comelico 39, 20135 Milano, Italy
besozzi@dico.unimi.it
[3] Università degli Studi di Bergamo, Dipartimento di Scienze della Persona Piazzale S. Agostino 2, 24129 Bergamo, Italy
paolo.cazzaniga@unibg.it
[4] Università degli Studi di Milano-Bicocca, Dipartimento di Statistica Via Bicocca degli Arcimboldi 8, 20126 Milano, Italy
dario.pescini@unimib.it

Abstract. Parameter estimation (PE) of biological systems is one of the most challenging problems in Systems Biology. Here we present a PE method that integrates particle swarm optimization (PSO) to estimate the value of kinetic constants, and a stochastic simulation algorithm to reconstruct the dynamics of the system. The fitness of candidate solutions, corresponding to vectors of reaction constants, is defined as the point-to-point distance between a simulated dynamics and a set of experimental measures, carried out using discrete-time sampling and various initial conditions. A multi-swarm PSO topology with different modalities of particles migration is used to account for the different laboratory conditions in which the experimental data are usually sampled. The whole method has been specifically designed and entirely executed on the GPU to provide a reduction of computational costs. We show the effectiveness of our method and discuss its performances on an enzymatic kinetics and a prokaryotic gene expression network.

1 Introduction

The emerging research area of Systems Biology aims at a better understanding of the dynamics of living cells in a quantitative way, by exploiting a synergistic integration between in silico analysis and wet experiments. In this context, a general limit to the effectiveness of computational modeling is the availability of physical parameters (e.g. reaction rates), which have a pivotal meaning in the regulation of biological systems, but that are usually hard or even impossible to measure. The problem of *parameter estimation* (PE) consists in the indirect determination of these unknown kinetic values, by

* Partially supported by Regione Lombardia, project "Network Enabled Drug Design (NEDD)".

M. Giacobini, L. Vanneschi, and W.S. Bush (Eds.): EvoBIO 2012, LNCS 7246, pp. 74–85, 2012.

exploiting the experimental data related to other quantities that can instead be measured by standard laboratory protocols (e.g. the abundance of some chemical species). Many methods for PE have been proposed, that rely either on some approximation strategy (see [3] and references therein) or to global optimization methods [10]. Following the latter line of research, in [1] we showed that *particle swarm optimization* (PSO) can outperform genetic algorithms for PE; indeed, this is an example of a dynamic optimization problem for which PSO can be efficiently applied [7].

PSO is a bio-inspired global optimization algorithm suitable for problems whose solutions are encoded as real-valued vectors [8]. PSO makes use of a population (the *swarm*) of candidate solutions (the *particles*), which are iteratively improved with respect to a fitness function. Each particle is identified by a position in the search space, and a velocity, that is used to update the current position of the particle at each iteration. The velocity, which is clamped to a maximum speed, changes according to two attractors: the (local) best position found by the particle itself, and the (global) best position found by any particle in the swarm. The influence of the attractors is weighted by random numbers, by cognitive and social factors to prevent the premature convergence to local minima, and by an inertia weight to avoid chaotic behaviors of the swarm.

In this work, we present a PSO-based method for PE that exploits the outcome of Gillespie's *stochastic simulation algorithm* (SSA) [5] to evaluate the fitness function. More precisely, the fitness is defined as the point-to-point distance between the measurements of the amount of some biochemical species, sampled during an experiment, and a simulated dynamics generated by SSA. In particular, one of the main novelties of our method in the context of PE is that it takes into account the quite common scenario of laboratory research, where multiple experiments are carried out starting from different initial conditions. Namely, our method is developed to deal with experimental data that consist of discrete-time temporal series of molecular species amounts, that are repeated a certain number of times. With respect to the approach previously presented in [1], this work considers a more realistic experimental target to perform the optimization, and a simplified definition of the fitness function to reduce the computational burden.

In this context, in order to estimate a common set of parameters able to reproduce the correct system dynamics in *all* conditions, we make use of a *multi-swarm* version of PSO where each swarm is assigned to a specific initial condition. Our method resembles the island-model of Evolutionary Computation [12], where a population of candidate solutions is partitioned into a set of disjoint sub-populations which evolve independently but interact by means of periodic migrations. In PSO terms, each sub-population corresponds to a swarm and the migration process is defined as the movement of particles between swarms. In this setting, a particle belonging to a particular swarm evaluates its own fitness by comparing the simulated dynamics with the experimental measures observed in that initial condition; then, to let the swarms cooperate, the global best particle of each swarm migrates toward another swarm at regular intervals of iterations, thus sharing the local estimates of the parameters values. Indeed, it is reasonable and biologically plausible to assume that there exists a common set of parameters that can simultaneously fit all the measures in all initial conditions, provided that the functioning of the biological system does not change in the chosen experimental settings, as it is presumed in the formulation of our problem. Finally, we have developed an

original implementation in CUDA of our multi-swarm PSO method embedding SSA, which is executed on general-purpose GPU (GPGPU) parallel architectures to provide a reduction of computational costs.

2 Parameter Estimation of Stochastic Biological Systems

In this section we propose a solution to the problem of PE for models of biological systems, defined according to the stochastic formulation of chemical kinetics [5]. We provide a formalization of the modeling framework and describe the type of experimental data that are assumed to be available as target for the estimation process. Then, we define a proper fitness function for this problem and explain the multi-swarm structure of PSO that will be exploited for PE in this context.

2.1 Formalization of the Problem

A stochastic model for a biological system can be defined by specifying the set $\mathcal{S} = \{S_1, \ldots, S_N\}$ of its molecular species, and a set $\mathcal{R} = \{R_1, \ldots, R_M\}$ of biochemical reactions which describe the interaction among the species [5]. Each reaction R_μ, $\mu = 1, \ldots, M$, is characterized by a stochastic kinetic constant $c_\mu \in \mathbb{R}^+$ (given in $time^{-1}$), a value that encompasses the physical and chemical properties of the reaction and that is generally hard or even impossible to measure experimentally. In the following, we assume that we have a complete knowledge of the sets $\mathcal{S}$ and $\mathcal{R}$, of the molecular amounts of the species initially present in the system, but no knowledge of the vector $\boldsymbol{\gamma}_c = (c_1, \ldots, c_M)$, whose components need to be estimated. The estimation process will be performed by relying on the following data:

- The system is analyzed under D different initial conditions, for some $D \geq 1$, that correspond to distinct experimental settings the system is exposed to, such as some chemical or environmental perturbation. In what follows we assume that different conditions are characterized by distinct initial amounts of some species in $\mathcal{S}$.
- For each initial condition, the experiment is repeated E times, for some $E \geq 1$ (usually, $E = 3$ in real laboratory experiments). This allows to account for the experimental errors in the measurement procedures, as well as for the variability of the system response that is due to the intrinsic biological noise.
- For each initial condition and for each replicate, the molecular amounts of a set $\mathcal{S}' \subseteq \mathcal{S}$ of molecular species are measured by means of standard laboratory technologies, where $\mathcal{S}' = \{S_1, \ldots, S_K\}$, for some $K \leq N$. Usually, N is in the order of tens, while K in the order of a few units for real systems.
- For each initial condition, for each replicate and for each species in $\mathcal{S}'$, the experimental measures are carried out at (a finite set of) time points $t_1, \ldots, t_F$, not necessarily sampled at regular intervals along the time course of the experiment.

We denote by $Y_k^{d,e}(t_f)$ the amount of species $S_k \in \mathcal{S}'$ measured at time t_f in the replicate e of initial condition d, where $k = 1, \ldots, K$, $f = 1, \ldots, F$, $e = 1, \ldots, E$, $d = 1, \ldots, D$. From now on, this set of measures will be called *discrete-time target series* (DTTS). Figure 1 (left side) shows an example of DTTS, with three replicates of

the measurements of some species S_k in a fixed experimental condition d. For simplicity, we assume that $Y_k^{d,e}(t_f) \in \mathbb{N}$, that is, it corresponds to the number of molecules of species S_k occurring in the system at time t_f. This is not restrictive, since a straightforward transformation of real-valued concentrations into a corresponding discrete number of molecules might be done as described, e.g., in [5].

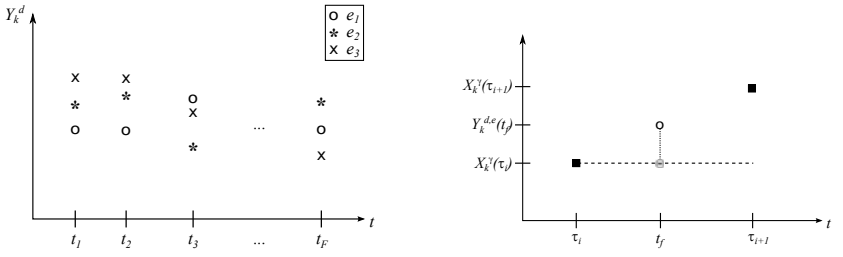


Fig. 1. *(Left)* Different discrete-time target series for species S_k in the experimental condition d, corresponding to three replicates e_1, e_2, e_3 of the same experiment. *(Right)* Identification of the amount of S_k, at the target time point t_f, exploiting an execution of SSA with parameters $\boldsymbol{\gamma}$.

In order to estimate the vector $\boldsymbol{\gamma_c}$ of reaction constants, we proceed by comparing two quantities: (i) the experimental DTTS of every $S_k \in \mathcal{S}'$, (ii) the molecular amounts of the same species S_k determined by executing in silico simulations of the dynamics of the stochastic model, which return the variation of the amounts of these species in time. To this aim, in this paper we exploit the stochastic simulation algorithm (SSA) [5], a seminal procedure used for reproducing the exact dynamics of biochemical systems, under the assumption that reactions take place inside a single volume where molecules are uniformly distributed. Given the state of the system at some time instant – that is, the set of molecular amounts of all the species in $\mathcal{S}$ – and a vector $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_M)$ of stochastic constants associated to the reactions in $\mathcal{R}$, SSA can simulate a correct dynamics of the system by computing, during each iteration, the length τ of the time interval required for the execution of a single reaction R_μ, $\mu = 1, \ldots, M$. So doing, SSA determines a set of consecutive time instants $\tau_0, \ldots, \tau_{max}$ – where τ_0 and τ_{max} are the fixed initial and last instants of the simulation – such that at the end of each step of length τ, with $\tau_0 \leq \tau \leq \tau_{max}$, the state of the system is instantly updated by deleting (adding, respectively) the molecules that appear as reagents (products, respectively) in reaction R_μ. The values of τ and μ are calculated by exploiting the propensity functions of the reactions [5], which are pseudo-probabilities computed step by step for each reaction R_μ by considering its stochastic constant c_μ and the molecular amounts of all the reagents of R_μ occurring in the system at the current step.

Let now $X_k^\gamma(\tau)$ denote the molecular amount of species $S_k \in \mathcal{S}'$ at time τ, with $k = 1, \ldots, K$ and $\tau_0 \leq \tau \leq \tau_{max}$, obtained by running SSA with some values $\gamma_1, \ldots, \gamma_M$ of the stochastic constants, as specified in an arbitrary vector $\boldsymbol{\gamma}$. In order to compare the simulated amount of species S_k and the measured amount of the same species in the DTTS, we need to determine the value $X_k^\gamma(t_f)$, that is, the amount of S_k taken in correspondence to each experimentally sampled time point t_f, $f = 1, \ldots, F$. This can

be done for each t_f by choosing two consecutive time instants $\tau_i, \tau_{i+1} \in [\tau_0, \tau_{max}]$ in a SSA simulation, such that $\tau_i \leq t_f \leq \tau_{i+1}$ and there exist no other τ', τ'' such that $\tau_i \leq \tau' \leq t_f \leq \tau'' \leq \tau_{i+1}$. Then, we can assume that $X_k^{\gamma}(t_f) = X_k^{\gamma}(\tau_i)$, since $X_k^{\gamma}(\tau) = X_k^{\gamma}(\tau_i)$ for every $\tau \in [\tau_i, \tau_{i+1})$ by definition of SSA (Figure 1, right side).

2.2 Fitness Function

Let $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_M)$ be a vector of arbitrary values of stochastic constants, with $\gamma_\mu \in \mathbb{R}^+$, $\mu = 1, \ldots, M$. Our aim is to determine the fitness of the parameter values specified in $\boldsymbol{\gamma}$ by comparing (i) the measured experimental data, and (ii) the result of a SSA execution using $\boldsymbol{\gamma}$. Namely, we want to establish if the simulated molecular amounts $X_k^{\gamma}(t_f)$ match the experimental measures $Y_k^{d,e}(t_f)$ of the DTTS, at corresponding time instants, for each species $S_k \in \mathcal{S}'$. To this aim, given a fixed initial condition $d \in \{1, \ldots, D\}$, for every species S_k and every $t_f \in \{t_1, \ldots, t_F\}$ we measure the point-to-point distance between $X_k^{\gamma}(t_f)$ and $Y_k^{d,e}(t_f)$, considering all E replicates carried out in the setting d. This leads to the following definition of the fitness function:

$$\mathcal{F}_d(\boldsymbol{\gamma}) = \frac{1}{E} \sum_{f=1}^{F} \sum_{k=1}^{K} \sum_{e=1}^{E} |Y_k^{d,e}(t_f) - X_k^{\gamma}(t_f)|. \tag{1}$$

The value $\mathcal{F}_d(\boldsymbol{\gamma})$ evaluates the quality[1] of the vector $\boldsymbol{\gamma}$ with respect to the DTTS in a fixed experimental setting d, by averaging over all the available E experimental data, and considering all the species in $\mathcal{S}'$ and all the sampled time instants t_f. In what follows, we denote by $\mathcal{F}_d(\boldsymbol{\gamma}_c)$ the fitness value obtained in condition d by using $\boldsymbol{\gamma}_c = (c_1, \ldots, c_M)$, that is, the correct parameter vector that we wish to estimate (this value will be used in Section 3.2 to identify a *successful run* of PSO). This fitness function is suitable to measure the quality of a chosen parameter vector in each experimental setting, independently from the other initial conditions that are considered in a laboratory to analyze the biological system. Since our aim is to face the problem of PE having at our disposal *a set* of D experimental DTTS, in the next section we show how to exploit a multi-swarm PSO to estimate a unique vector of parameters $\boldsymbol{\gamma}$ that can simultaneously fit all DTTS in all conditions. In other terms, we exploit a multi-swarm architecture to deal with multiple fitness-cases – distributing one case per swarm – and take advantage of particles migration to achieve a unique result for all these cases, as the PE problem we are considering here actually requires.

In addition to the fitness function given in Equation 1, we have also tested the strategy of *fitness sharing* [6], in order to mitigate the premature convergence of solutions, a problem typical of population-based optimization heuristics. Within a swarm, fitness sharing modifies the fitness value of each particle, in such a way that the more diverse are advantaged. To this aim, $\mathcal{F}_d(\boldsymbol{\gamma})$ has been modified as follows:

$$\mathcal{F}_{sh}(\boldsymbol{\gamma}) = \mathcal{F}_d(\boldsymbol{\gamma}) \cdot \frac{\max\{\|\boldsymbol{\gamma}_i - \boldsymbol{\gamma}_j\| \mid i \neq j\}}{\sum_{i \neq j} \|\boldsymbol{\gamma}_i - \boldsymbol{\gamma}_j\|} \tag{2}$$

where $\|\boldsymbol{\gamma}_i - \boldsymbol{\gamma}_j\|$ denotes the Euclidean norm of particles $\boldsymbol{\gamma}_i$ and $\boldsymbol{\gamma}_j$ of the same swarm.

[1] We stress the fact that by exploiting SSA to generate the $X_k^{\gamma}(t_f)$ values for the fitness calculation, it is very unlike to reach the (ideal) value of zero, due to the intrinsic stochasticity.

2.3 Multi-swarm PSO

Our multi-swarm topology of PSO is structured as follows. First, we consider a swarm σ_d for each experimental setting $d = 1, \ldots, D$, consisting of n particles $\boldsymbol{\gamma}_d$ corresponding to vectors of values for the stochastic constants, that is, $\boldsymbol{\gamma}_d = (\gamma_{d,1}, \ldots, \gamma_{d,M})$, with $\gamma_{d,\mu} \in \mathbb{R}^+$, $\mu = 1, \ldots, M$. As described in Section 2.1, this vector is used to execute the SSA to generate the sets of values $X_k^{\gamma}(t_f)$ associated to this particle, for all $S_k \in \mathcal{S}'$. The fitness of particle $\boldsymbol{\gamma}_d$ is then evaluated according to Equation 1 by using the experimental data of condition d, $Y_k^{d,e}(t_f)$. So doing, each swarm performs a PE independently from the others, and determines its global best particle, denoted $\boldsymbol{\gamma}_d^{best}$, which is the vector of stochastic constants that matches the dynamics of the system in condition d in the best possible way.

Then, in order to estimate a set of stochastic constants that is common for all swarms, and that is able to reproduce the correct system dynamics in *all* conditions, we exploit the *migration* of particles [11], which is used here to share the global best particle of each swarm among the other swarms. The migration takes place at regular intervals, that is, after a number IT_{mig} of iterations, with $1 \leq IT_{mig} \leq IT_{max}$, where IT_{max} is the maximum number of iterations of the PSO. Two migration topologies have been considered. In the *static topology* (ST), swarms are arranged in a unidirectional ring, as also considered in [11], whereby swarm σ_d sends its global best particle to swarm $\sigma_{(d \bmod D)+1}$. In the *dynamic topology* (DT), the interconnections among swarms are chosen randomly and updated at each step of migration, provided that each swarm always sends only its global best particle and receives only one global best particle from some other swarm. In the DT, if a swarm remains isolated at some migration step, then it will neither receive nor send particles to other swarms at that step. Chosen either the ST or DT, the migration acts by removing the worst particle of the swarm and replacing it with the incoming particle, in order to maintain the swarm size.

Note that the notion of DT for migration was first introduced for parallel genetic algorithms [9], where the migration topology is dynamically updated according to some properties of the populations; in our method, instead, the topology is randomly updated.

3 Implementation and Discussion of the Methodology

In this section we briefly present the architectural choices behind the GPU-based implementation of our PE methodology, and then we present the results obtained on two biological systems, a basic scheme of enzymatic kinetics and a prokaryotic gene expression network.

3.1 A GPU-Based Multi-swarm PSO

The methodology described so far is computationally expensive, because of the huge number of fitness evaluations performed by the multi-swarm PSO. Being the PSO an inherently parallel algorithm, we improved the performances of our PE method by using the GPGPU architecture, that exploits the great computational power of modern video cards. Our method has been implemented with NVIDIA's CUDA, a GPGPU computing library combining multi-threading with the *single instruction multiple data* architecture.

In this implementation, a single thread is executed for each particle. Multiple threads are organized in multi-dimensional structures called *blocks*; threads belonging to the same block can cooperate by using a small amount of *shared memory*. We associate each swarm to a different block, so that parallel algorithms can exploit this feature (e.g. *parallel reduction* to find the worst and best particles of each swarm). Moreover, fitness evaluations are independent and can be computed in parallel, by simultaneously executing all SSA simulations in separate threads. Our implementation of PSO is synchronous: the D vectors of the best particles $\boldsymbol{\gamma}_d^{best}$ are updated when all SSA simulations are concluded. Every IT_{mig} iterations, the synchronization is followed by the migration of particles, according to the static or the dynamically-generated topology, as described in Section 2.3. The dynamical generation of a new topology is done efficiently, in-place and in linear time with Durstenfeld's algorithm [4].

3.2 Results

In this section we present the results of our method for the estimation of the stochastic constants of two simple systems: the *Michaelis-Menten kinetics* (MM) and a *prokaryotic auto-regulatory gene network* (PGN). For these systems, the DTTS of every experiment replicate e has been generated in silico by first averaging the amount of target species over 1000 SSA executions, each one performed using the vector of correct parameters $\boldsymbol{\gamma_c}$, and then sampling the outcome at $F = 10$ time instants for each species.

The MM system describes the catalytic transformation of a substrate (S) into a final product (P) mediated by the activity of an enzyme (E), passing through the reversible formation of an intermediate complex (ES) [1]. The chemical reactions corresponding to MM are: $R_1 : E + S \xrightarrow{c_1} ES, R_2 : ES \xrightarrow{c_2} E + S, R_3 : ES \xrightarrow{c_3} E + P$, where $\boldsymbol{\gamma}_c = (2.5 \cdot 10^{-3}, 0.1, 5)$. For MM we consider $E = 3$ replicates for each of the $D = 4$ distinct experimental settings, which are characterized by different initial amounts of molecules for the substrate and the enzyme (S, E): $\sigma_1 : (1000, 750)$, $\sigma_2 : (2000, 750)$, $\sigma_3 : (500, 750)$, $\sigma_4 : (1000, 500)$.

The PGN system is a simple example of auto-regulation mechanism of gene expression, whereby a gene (DNA) that codes for a protein (P) is inhibited by the binding with a dimer of the protein itself ($DNA{\cdot}P_2$) [13]. The chemical reactions corresponding to PGN are: $R_1 : DNA + P_2 \xrightarrow{c_1} DNA \cdot P_2$, $R_2 : DNA \cdot P_2 \xrightarrow{c_2} DNA + P_2$, $R_3 : DNA \xrightarrow{c_3} DNA + mRNA$, $R_4 : mRNA \xrightarrow{c_4} \lambda$, $R_5 : 2P \xrightarrow{c_5} P_2$, $R_6 : P_2 \xrightarrow{c_6} 2P$, $R_7 : mRNA \xrightarrow{c_7} mRNA + P$, $R_8 : P \xrightarrow{c_8} \lambda$, where λ denotes a degradation and $\boldsymbol{\gamma}_c = (0.1, 0.7, 0.35, 0.3, 0.1, 0.9, 0.2, 0.1)$. For PGN we consider $E = 3$ replicates for each of the $D = 3$ distinct experimental settings, which are characterized by different initial amounts of DNA molecules: $\sigma_1 : (50)$, $\sigma_2 : (100)$, $\sigma_3 : (1000)$.

Several preliminary tests have been executed to find the best setting for our multi-swarm PSO, that has been used to generate the results presented below: all PEs have been executed 50 times under the same conditions; the maximum number of iterations is $IT_{max} = 200$; the size of each swarm has been set to $n = 32$ particles; the interval for the migration of particles is $IT_{mig} = 10$ iterations; the cognitive and social factors of particles have been set to 1.9; the inertia weight is equal to 0.9; the maximum velocity of particles is limited to $1/10$ of the maximum distance between solutions.

The aim of the first analysis we performed was to determine the effect of combining fitness sharing (FS) with the two migration strategies (ST and DT). We run four PEs of MM corresponding to the following cases: (1) FS-ST, (2) FS-DT, (3) noFS-ST, (4) noFS-DT. In cases (1) and (2), the fitness was evaluated as defined in Equation 2, while in cases (3) and (4) as in Equation 1. The results of this analysis indicate that fitness sharing does not provide a better estimation of the parameters (data not shown), while the comparison between ST and DT migration suggests that the latter improves the optimization process. As a matter of fact, the particles with the smallest fitness values have been identified when particles migrate according to DT (Figure 2, left side). Hence, all the analysis of the two biological systems have been executed considering case (4).

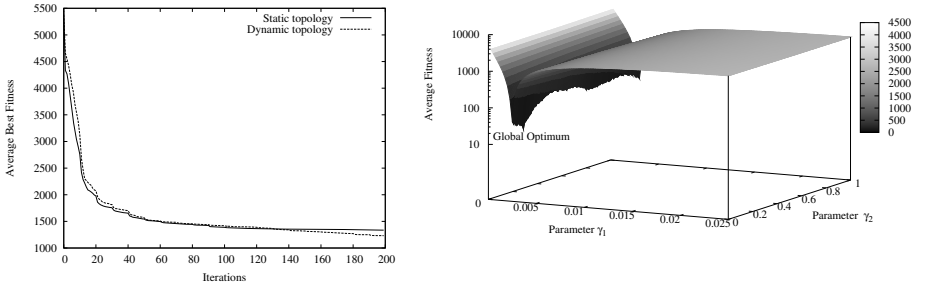


Fig. 2. (*Left*) Comparison between the average best fitness of the MM optimization with static (solid line) and dynamic topology (dashed line). (*Right*) Fitness landscape of MM considering the variation of parameters γ_1 and γ_2.

To determine how much the estimated values of stochastic constants of a particle $\boldsymbol{\gamma}$ are close to the correct parameter vector $\boldsymbol{\gamma}_c$, we define the *mean error* $\overline{\varepsilon}_\gamma$:

$$\overline{\varepsilon}_\gamma = \frac{1}{M}\sum_{\mu=1}^{M} \varepsilon_{\gamma_\mu} = \frac{1}{M}\sum_{\mu=1}^{M} \frac{|c_\mu - \gamma_\mu|}{c_\mu}, \tag{3}$$

where γ_μ $(\mu = 1, \ldots, M)$ denotes the value of each stochastic constant codified in the particles of PSO, and ε_{γ_μ} is the relative error of the constant γ_μ with respect to the real constant c_μ [13]. The best solution found by the multi-swarm PSO is denoted as $\boldsymbol{\gamma}^* = \{\boldsymbol{\gamma} \mid \overline{\varepsilon}_{\gamma^*} = \min\{\overline{\varepsilon}_\gamma \mid \boldsymbol{\gamma} \in \bigcup_d \sigma_d\}\}$, that is, we select the particle having the minimum mean error value among all swarms. We also compute the *average mean error* $\langle \overline{\varepsilon}_{\gamma^{sr}} \rangle$, that is calculated averaging the values of the mean errors $\overline{\varepsilon}_{\gamma^{sr}}$, which denote the errors of the best particles found in a *successful run*. A run is considered successful if there is a best particle $\boldsymbol{\gamma}^{sr}$ of some swarm σ_d having a fitness value $\mathcal{F}_d(\boldsymbol{\gamma}^{sr}) \leq 1.1\mathcal{F}_d(\boldsymbol{\gamma}_c)$.

We started by analyzing the influence that specific stochastic constants can have on the estimation process. To this aim, we executed our PE method by fixing to c_μ the value of one or more components γ_μ of particles $\boldsymbol{\gamma}$. In Table 1 we present the derived values of the mean error $\overline{\varepsilon}_{\gamma^*}$: when we estimate only a subset of components of $\boldsymbol{\gamma}$ (from column 2 to 7), we obtain better solutions. In particular, if we look at the average mean error

$\langle \overline{\varepsilon}_{\gamma^{sr}} \rangle$ (second line), we also note that the PEs involving the optimization of γ_2 always lead to the highest errors. This result is even clearer considering the single contribution of each γ_μ^* to the mean error of the best particle $\boldsymbol{\gamma}^*$ (last three lines of Table 1).

Table 1. Error of the best particles $\boldsymbol{\gamma}^*$ and successful runs particles $\boldsymbol{\gamma}^{sr}$ of MM: influence of the components γ_1, γ_2, γ_3 on the PE process

Constants	$\gamma_1, \gamma_2, \gamma_3$	γ_1, γ_2, c_3	c_1, γ_2, γ_3	γ_1, c_2, γ_3	γ_1, c_2, c_3	c_1, γ_2, c_3	c_1, c_2, γ_3
$\overline{\varepsilon}_{\gamma^*}$	$2.51 \cdot 10^{-1}$	$6.07 \cdot 10^{-2}$	$7.66 \cdot 10^{-3}$	$4.77 \cdot 10^{-3}$	$2.00 \cdot 10^{-5}$	$2.95 \cdot 10^{-2}$	$2.13 \cdot 10^{-3}$
$\langle \overline{\varepsilon}_{\gamma^{sr}} \rangle$	1.56	$8.52 \cdot 10^{-1}$	$2.58 \cdot 10^{-1}$	$2.21 \cdot 10^{-2}$	$4.21 \cdot 10^{-3}$	$2.80 \cdot 10^{-1}$	$1.30 \cdot 10^{-2}$
$\varepsilon_{\gamma_1}/\overline{\varepsilon}_{\gamma^*}$	$1.39 \cdot 10^{-1}$	$2.62 \cdot 10^{-2}$	0.00	$8.08 \cdot 10^{-1}$	1.00	0.00	0.00
$\varepsilon_{\gamma_2}/\overline{\varepsilon}_{\gamma^*}$	$7.45 \cdot 10^{-1}$	$9.74 \cdot 10^{-1}$	$7.57 \cdot 10^{-1}$	0.00	0.00	1.00	0.00
$\varepsilon_{\gamma_3}/\overline{\varepsilon}_{\gamma^*}$	$1.16 \cdot 10^{-1}$	0.00	$2.43 \cdot 10^{-1}$	$1.92 \cdot 10^{-1}$	0.00	0.00	1.00

Table 2. Error of the best particles $\boldsymbol{\gamma}^*$ and successful runs particles $\boldsymbol{\gamma}^{sr}$ of MM: influence of the number of target species considered in DTTS on the PE process

Species	S, E, ES, P	S, E, ES	E, ES, P	S, E	S, ES	S, P
$\overline{\varepsilon}_{\gamma^*}$	$2.51 \cdot 10^{-1}$	$3.50 \cdot 10^{-1}$	$4.12 \cdot 10^{-2}$	$3.38 \cdot 10^{-1}$	$7.31 \cdot 10^{-2}$	$1.05 \cdot 10^{-1}$
$\langle \overline{\varepsilon}_{\gamma^{sr}} \rangle$	1.56	11.1	13.8	13.7	8.76	6.23
$\varepsilon_{\gamma_1}/\overline{\varepsilon}_{\gamma^*}$	$1.39 \cdot 10^{-1}$	$8.71 \cdot 10^{-3}$	$2.32 \cdot 10^{-1}$	$5.57 \cdot 10^{-3}$	$1.82 \cdot 10^{-1}$	$5.52 \cdot 10^{-2}$
$\varepsilon_{\gamma_2}/\overline{\varepsilon}_{\gamma^*}$	$7.45 \cdot 10^{-1}$	$9.64 \cdot 10^{-1}$	$4.89 \cdot 10^{-1}$	$9.87 \cdot 10^{-1}$	$8.06 \cdot 10^{-1}$	$7.45 \cdot 10^{-1}$
$\varepsilon_{\gamma_3}/\overline{\varepsilon}_{\gamma^*}$	$1.16 \cdot 10^{-1}$	$2.75 \cdot 10^{-2}$	$2.80 \cdot 10^{-1}$	$7.15 \cdot 10^{-3}$	$1.13 \cdot 10^{-2}$	$2.00 \cdot 10^{-1}$

To better understand this result, we plot in Figure 2 (right) the fitness landscape of MM considering the variation of parameters γ_1 and γ_2. Whereas parameter γ_1 induces a unique (global) minimum on the fitness landscape, γ_2 features several local minima which mislead the particles. The figure displays another interesting characteristic of MM system: as the values of γ_1 increases, the fitness landscape shows a "flat" portion (characterized by many local optima, not clearly visible in this graphical representation) where particles can get stuck, thus resulting in solutions with a poor quality.

Afterwards, we compared the results obtained when using, from time to time, the DTTS of one or more molecular species of MM. If we assume the availability of the experimental measures for only a few species, that is, less information during the optimization process, then the values of the average mean error increase. On the other hand, the error of the best particles does not significantly change from case to case, proving the efficiency of our method. For space limits, we report in Table 2 only the results obtained on a few subsets of $\mathcal{S}$ (similar results were obtained in all the other cases).

In general, since the correct values of $\boldsymbol{\gamma}_c$ are not known, the mean error $\overline{\varepsilon}_\gamma$ of the particles $\boldsymbol{\gamma}$ cannot be computed to assess the quality of the estimated stochastic constants. To this aim, a more reliable and suitable way to proceed is to compare the experimental data used as target and the simulated dynamics of the system, generated using the set of stochastic constants of the best particles. Therefore, for each swarm σ_d we select the best particle $\boldsymbol{\gamma}_d^{best}$, we perform the stochastic simulations by using the D corresponding parameter vectors, and then we choose the solution that matches all the DTTS in the best qualitative and quantitative way.

In the case of MM system, the best solution found is $\gamma^* = (0.00236337, 0.119315, 5.18206)$. In Figure 3 we plot the dynamics (solid lines) obtained by considering the $D = 4$ different initial conditions stated above and the target DTTS (dots) used to compute the fitness values. As clearly shown in this figure, the dynamics of all species generated with the solution γ^* generated by the multi-swarm PSO match very closely the targets for all initial conditions.

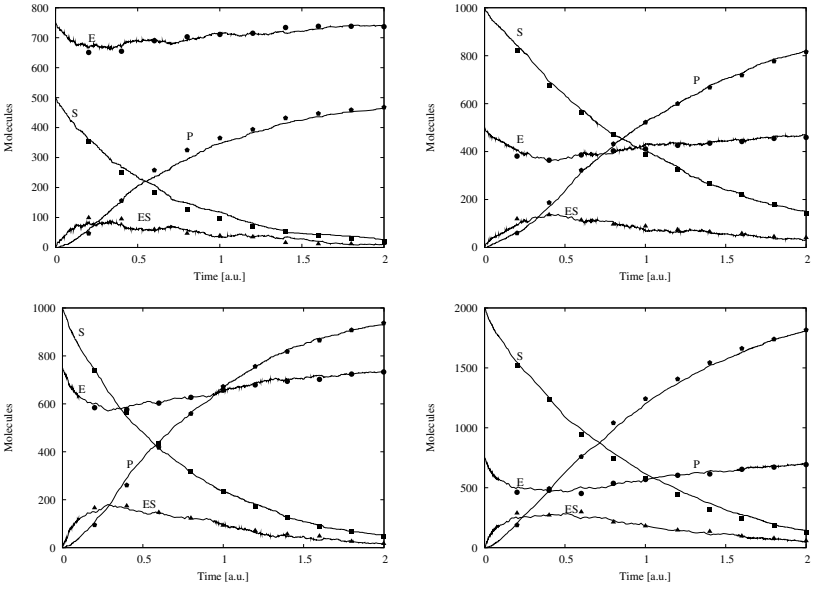


Fig. 3. Dynamics of MM obtained using the constants found by the best solution generated by the four swarms of PSO (solid lines) compared to the DTTS (dots), under different initial conditions (for the sake of clarity, only one DTTS per species is shown)

Finally, in Figure 4 we show the comparison between the simulations of PGN performed using the best solution found by the multi-swarm PSO – which corresponds to $\gamma^* = (0.176459, 1, 0.384382, 0.314425, 0.106459, 1, 0.184306, 0.0413758)$ – and the DTTS used during the optimization. In particular, the plots report the dynamics of only two molecular species of PGN, namely DNA and $mRNA$, where the initial conditions correspond to swarms σ_2 (top graphics) and σ_3 (bottom graphics), as given above. Once more, the dynamics match very closely the targets for both initial conditions; this is true also for the other species and for the initial condition of swarm σ_1 (data not shown).

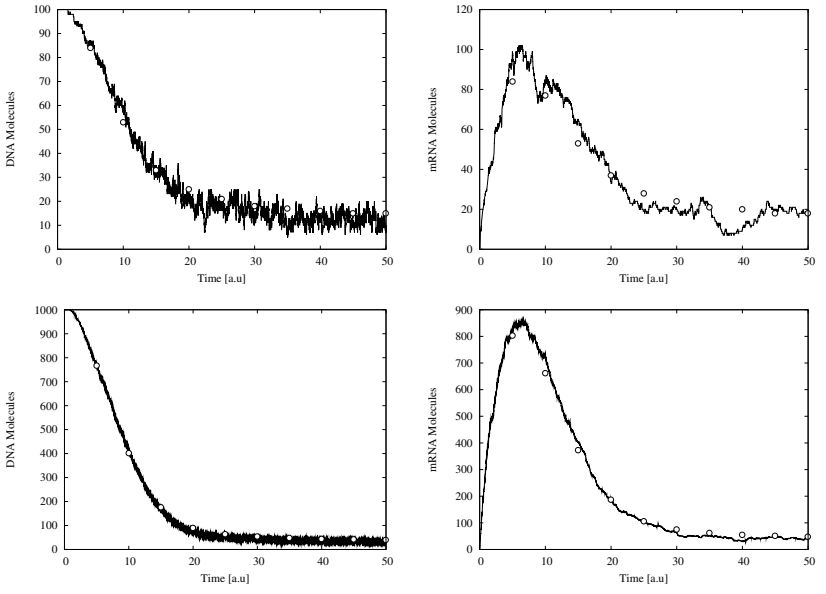


Fig. 4. Dynamics of PGN obtained using the constants found by the best solution generated by the three swarms of PSO (solid lines) compared to the target used during the optimization (dots), under different initial conditions (for the sake of clarity, only one DTTS per species is shown)

4 Conclusions

In this paper we have proposed a method for the estimation of reaction constants in stochastic biological systems, which exploits experimental discrete-time target series. The foremost novelties of our method, in the context of parameter estimation, rely on several peculiar features. First of all, it does not require a uniform sampling rate, nor observations of every chemical species in the system. It can handle experimental samples coming from multiple experiments executed under different initial conditions, and produces a single estimation for all conditions exploiting a multi-swarm version of PSO, where the populations converge to a common solution by periodically exchanging their best particles. The fitness function we use is a point-to-point distance between the experimental samples and the dynamics simulated with a stochastic algorithm, which allows to account for the effects of biological noise. Moreover, the method is developed using a GPGPU computing architecture, to exploit the intrinsic parallelism of PSO. This implementation works entirely on the GPU, running a separate thread for each particle and achieving a boost ($24\times$, according to our tests) with respect to a strictly sequential implementation. For instance, for the computation of 1280000 fitness values, the execution of the GPU implementation on a workstation with a Tesla C1060 takes 14 minutes, while the sequential version takes 346 minutes, executed on a CPU Intel Core Duo 6700 (2.66 GHz).

By profiling the computational costs of our method, we have evidenced that SSA is responsible for the largest part of the effort. Therefore, a further improvement of our method will be focused on the implementation of a less computationally expensive

simulation algorithm, like the tau-leaping [2]. This will also allow us to improve our method with respect to the following aspects. First, we might extend the fitness function defined here, that has been intentionally simplified with respect to [1] to reduce the computational burden. Second, in our implementation every fitness is calculated by comparing the experimental samples against the dynamics produced by a single simulation; though, due to the stochasticity of biological systems, the noise of one simulated dynamics might mislead the estimation. Thus, by exploiting a faster stochastic simulation algorithm we will be able to compare the experimental data against the averaged dynamics of *many* different simulations. Third, the reduction of the computational costs related to the stochastic simulations might allow us to perform a larger number of iterations, potentially converging to better solutions.

As a final remark, we highlight that this GPU-based method for PE has been developed to the aim of analyzing large biological systems of great biological interest, consisting of many reactions and many species, which are already under multidisciplinary investigations in our research group.

References

1. Besozzi, D., Cazzaniga, P., Mauri, G., Pescini, D., Vanneschi, L.: A comparison of Genetic Algorithms and Particle Swarm Optimization for parameter estimation in stochastic biochemical systems. In: Pizzuti, C., Ritchie, M.D., Giacobini, M. (eds.) EvoBIO 2009. LNCS, vol. 5483, pp. 116–127. Springer, Heidelberg (2009)
2. Cao, Y., Gillespie, D.T., Petzold, L.: Efficient step size selection for the tau-leaping simulation method. J. Chem. Phys. 124, 44109 (2006)
3. Chou, I.C., Voit, E.O.: Recent developments in parameter estimation and structure identification of biochemical and genomic systems. Math. Biosci. 219, 57–83 (2009)
4. Durstenfeld, R.: Algorithm 235: Random permutation. Commun. ACM 7, 420 (1964)
5. Gillespie, D.T.: Exact stochastic simulation of coupled chemical reactions. J. Comp. Phys. 81, 2340–2361 (1977)
6. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley (1989)
7. Janson, S., Middendorf, M.: A hierarchical particle swarm optimizer for dynamic optimization problems. In: Raidl, G.R., Cagnoni, S., Branke, J., Corne, D.W., Drechsler, R., Jin, Y., Johnson, C.G., Machado, P., Marchiori, E., Rothlauf, F., Smith, G.D., Squillero, G. (eds.) EvoWorkshops 2004. LNCS, vol. 3005, pp. 513–524. Springer, Heidelberg (2004)
8. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proc. of the IEEE International Conference on Neural Networks, Piscataway, NJ, vol. IV, pp. 1942–1948 (1995)
9. Lin, S.C., Punch III, W.F., Goodman, E.D.: Coarse-grain parallel genetic algorithms: Categorization and new approach. In: Proc. of Sixth IEEE Symposium on Parallel and Distributed Processing, pp. 28–37. IEEE Computer Society Press (1994)
10. Moles, C.G., Mendes, P., Banga, J.R.: Parameter estimation in biochemical pathways: a comparison of global optimization methods. Genome Res. 13, 2467–2474 (2003)
11. Romero, J., Cotta, C.: Optimization by island-structured decentralized particle swarms. In: Reusch, B. (ed.) Computational Intelligence, Theory and Applications. AISC, vol. 33, pp. 25–33. Springer, Heidelberg (2005)
12. Tanese, R.: Distributed genetic algorithms. In: Proc. of Third Int. Conference on Genetic Algorithms, pp. 434–439. Morgan Kaufmann Publishers, San Francisco (1989)
13. Wang, Y., Christley, S., Mjolsness, E., Xie, X.: Parameter inference for discretely observed stochastic kinetic models using stochastic gradient descent. BMC Syst. Biol. 4(1), 99 (2010)

Tracking the Evolution of Cooperation in Complex Networked Populations

Flávio L. Pinheiro[1], Francisco C. Santos[1,2], and Jorge M. Pacheco[1,3]

[1] ATP-Group, CMAF, Instituto para a Investigacao Interdisciplinar, P-1649-003 Lisboa Codex, Portugal
[2] Departamento de Engenharia Informática, Instituto Superior Técnico, Universidade Técnica de Lisboa, Av. Rovisco Pais, 1, 1049-001 Lisboa, Portugal
[3] Departamento de Matemática e Aplicações, Universidade do Minho, 4710 - 057 Braga, Portugal

Abstract. Social networks affect in such a fundamental way the dynamics of the population they support that the global, population-wide behavior that one observes often bears no relation to the agent processes it stems from. Up to now, linking the global networked dynamics to such agent mechanisms has remained elusive. Here we define an observable dynamic and use it to track the self-organization of cooperators when co-evolving with defectors in networked populations interacting via a Prisoner's Dilemma. Computations on homogeneous networks evolve towards the coexistence between cooperator and defector agents, while computations in heterogeneous networks lead to the coordination between them. We show how the global dynamics co-evolves with the motifs of cooperator agents in the population, the overall emergence of cooperation depending sensitively on this co-evolution.

Keywords: Complex Networks, Self-Organization, Cooperation, Evolutionary Game Theory, Evolutionary Dynamics.

1 Introduction

Dynamical processes involving populations of agents constitute paradigmatic examples of complex systems. From epidemic outbreaks to opinion formation, evolutionary and learning behavioral dynamics, the impact of the underlying web of ties in the overall behavior of the population is well known [1,6,10,12,13,15,21,22,26,30,41,42]. Furthermore, it is often impossible to avoid such structures when applications require the deployment of agents under physical or other constraints as it is with network routing [18,29], computational intelligence techniques [7,8,43] and sensor networks [2].

In this context, Evolutionary Games [35] provides one of the most sophisticated examples of complex system dynamics in which the role of the underlying network topology proves ubiquitous. For instance, when cooperation is modeled as a Prisoner's dilemma game (**PD**), cooperation may emerge (or not) depending on how agents are networked [14,16,23–25,27,31,32,37,38]. Up to now, multi-agent based models were unable to identify the detailed mechanism by which

M. Giacobini, L. Vanneschi, and W.S. Bush (Eds.): EvoBIO 2012, LNCS 7246, pp. 86–96, 2012.

local self-regarding actions lead to a collective cooperative scenario, in particular relating it to the network topology. In the following, we devise a means to establish such a link between individual and collective behaviors, in terms of the underlying network topology. To this end we make use of evolutionary game dynamics, although the method should be easily applicable to other dynamical processes taking place on general complex networks.

2 Results and Discussion

2.1 Evolution of Cooperation in Finite Well-Mixed Populations

Let us consider two agents who can each adopt one of two possible behaviors: *Cooperator* (C) or *Defector* (D). Whenever they interact, four outcomes are possible: Two Cs receive R (reward) each, whereas each receives P (punishment) if both are Ds. Whenever a C interacts with a D, the C gets S (the sucker's payoff) whereas the D gets T (temptation to defect). These outcomes can be summarized through the so-called payoff matrix,

$$\begin{array}{c} \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cc} C & D \end{array} \\ \begin{pmatrix} R & S \\ T & P \end{pmatrix} \end{array} \tag{1}$$

Whenever $T > R > P > S$ one obtains the **PD** [4,35]. For simplicity, we formalize the **PD** game in terms of a single parameter B (benefit) by defining $T = B > 1$, $R = 1$, $S = 1 - B$ and $P = 0$.

In the context of Evolutionary Game Theory [35], the payoff of an agent is associated with her/his fitness that is her/his social success. Thus, behaviors that provide higher rewards are imitated more frequently and spread in the population. Here, evolution and strategy update is modelled via a stochastic birth-death process in finite populations of size N, often referred as *pairwise comparison* rule [35,39]. At each iteration, a randomly selected agent x adopts the strategy of a randomly selected neighbor y with probability given by the Fermi distribution

$$p = [1 + e^{-\beta(f_y - f_x)}]^{-1}, \tag{2}$$

where the fitness values f_x (f_y) stand for the accumulated payoff of x (y) and β controls the intensity of selection measuring the importance of the agent payoffs and stochastic effects in the imitation process [39].

In the limit of well-mixed populations of size N – where agents may interact with any other agent in the population –, Cs are always worse off than Ds, and will be outcompeted under natural selection [35]. Mathematically, this means that the gradient of selection [28,34,39]

$$G(j) = T^+(j) - T^-(j) \tag{3}$$

is negative for all j, where j stands for the number of Cs in the population and

$$T^{\pm}(j) = \frac{N-j}{N}\frac{j}{N}\frac{1}{1 + e^{\pm\beta(f_D - f_C)}} \tag{4}$$

represent the probabilities to increase/decrease the number of Cs in the population [40].

The elegance of this result (despite the doomsday scenario for Cs) is best appreciated when realizing that the population ends up adopting the Nash-equilibrium of a **PD** game interaction between two agents: everybody defects. Consequently, there is no difference in the outcome of the game, from an agent or from a (collective) population wide perspective. This result holds in structureless populations, a feature which is seldom observed in practice, with strong implications in many natural phenomena.

It is noteworthy that the general methodology discussed in the next section is independent from the stochastic update rule adopted in the evolutionary process. Moreover, this stochastic update is more general one could initially foresee, as the ensuing dynamics may be also shown to be equivalent to the replicator equation [17,40] and to finite action learning automata in the limit of infinite, well-mixed populations [9,36,41].

2.2 Gradients of Selection in Structured Populations

A homogeneous network, in which all agents engage in the same number of games (k) with their first neighbors, represents the simplest case of a structured population, where agents occupy the nodes of the network, whose links determine who is neighbor of whom. Unlike well-mixed populations, even in such simple homogeneous scenario where all agents share the same number of neighbors, agents with the same strategy no longer necessarily share the same fitness (here associated with game payoff): fitness becomes context-dependent and so does the gradient of selection, which is now impossible to compute analytically.

To overcome this problem, we define the Average Gradient of Selection (***AGoS***), denoting it by $G^A(j)$ as the average i) over all possible transitions taking place in every node of the network throughout evolution, and ii) over a large number of networked evolutions. For each agent i we compute the probability of changing behavior at time t,

$$T_i = \frac{1}{k_i} \sum_{m=1}^{\bar{n_i}} [1 + e^{-\beta(f_m - f_i)}]^{-1}, \tag{5}$$

where k_i stands for the degree of node i and $\bar{n_i}$ for the number of neighbors of i having a strategy different from that of i. The ***AGoS*** at a given time t of simulation p, where we have j Cs, is defined as,

$$G_p(j,t) = T_A^+ - T_A^- \tag{6}$$

where, $T_A^{\pm} = \frac{1}{N} \sum_{i=1}^{AllCs}{}^{AllDs} T_i(t)$.

For a given network type, we run $\Omega = 2 \times 10^7$ simulations (using 10^3 randomly generated networks) starting from all possible initial fractions j/N of Cs. Each configuration of the population is associated with the fraction j/N of Cs. Evolutions run for $\Delta = 10^5$ time steps. Hence, the overall, time-independent

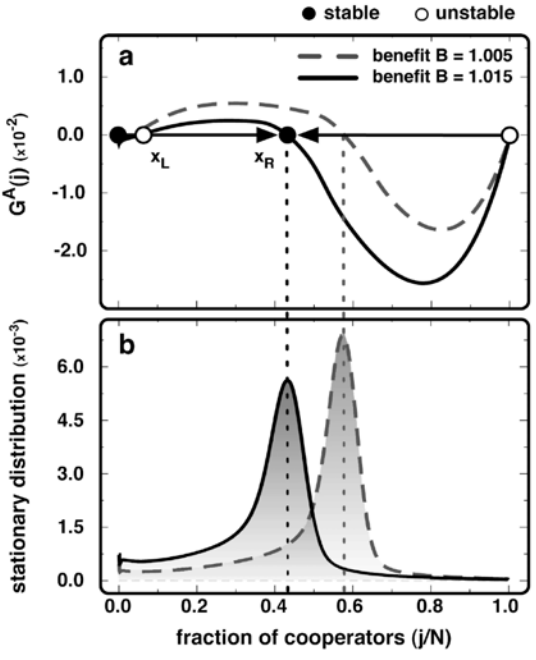


Fig. 1. Time-independent *AGoS*. **(a)** We plot $G^A(j)$ for a population of players interacting via a **PD** in a homogeneous random network, for two values of the benefit B. Globally, $G^A(j)$ indicates that the population evolves towards a co-existence scenario. **(b)** Stationary distributions showing the pervasiveness of each fraction j/N in time. In line with the ***AGoS*** in a), the population spends most of the time in the vicinity of the stable-like root x_R of $G^A(j)$. When $j/N \approx 0$, Cs become disadvantageous, giving rise to an unstable-like root x_L of $G^A(j)$ which, however, plays a minor role as shown ($N = 10^3$, $k = 4$ and $\beta = 1.0$). Homogeneous random networks were obtained by repeatedly swapping the ends of pairs of randomly chosen links of a regular lattice [33].

AGoS is given by the average

$$G^A(j) = \frac{1}{\Omega\Delta} \sum_{t=1}^{\Delta} \sum_{p=1}^{\Omega} G_p(j,t) \tag{7}$$

over all simulations and time-steps.

The gradient of selection in networks has to be computed numerically and has the nice property of being network dependent but context independent, as it recovers a population most likely direction of selection. As demonstrated below, ***AGoS*** allow us to follow in time the evolutionary dynamics from a global, population-wide perspective, as opposed to an agent perspective, which can always be inferred from the structure of the payoff matrix.

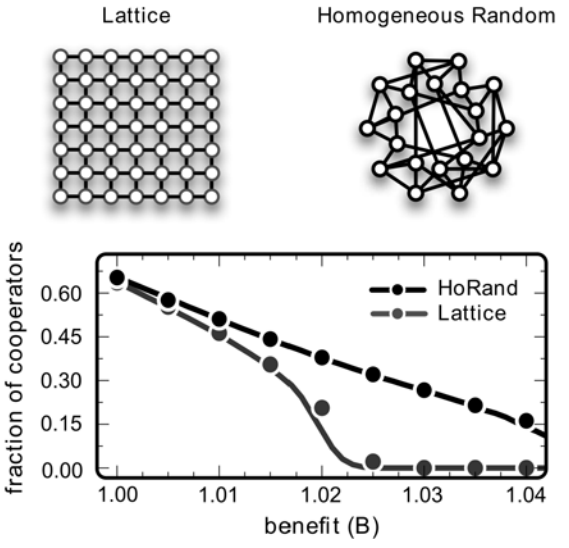


Fig. 2. Evolutionary dynamics cooperation in homogeneous networks. We plot the interior roots x_R of $G^A(j)$ (circles and squares) for a **PD** ($T = B$, $R = 1$, $P = 0$, $S = 1 - B$) in homogeneous networks, from random networks (circles) to ordered lattices (squares), as a function of the benefit B. $G^A(j)$ indicates that the population evolves towards a stationary fraction x_R of Cs. This is confirmed by the stationary states (lines) obtained via computer simulations starting from 50% of Cs and Ds randomly placed in each network. ($N = 10^3$, $k = 4$ and $\beta = 0.1$).

2.3 Results for Homogeneous Networks

The results for $G^A(j)$ on homogeneous random networks are shown in Fig. 1a. Unlike well-mixed populations, where cooperation has no chance and $G^A(j) < 0$ for all values of j, homogeneous networks can sustain cooperation [24, 33, 37]. The shape of $G^A(j)$ suggests that, even though every agent engages in a **PD**, from a global, population-wide perspective, homogeneous networks give rise to an emerging collective dynamics promoting the co-existence between Cs and Ds defined by a co-existence point at $j/N = x_R$.

This hypothesis is confirmed when one computes the stationary distribution, which measures the fraction of time that the population spends in each available state j/N before reaching fixation (Fig. 1b). It represents the pervasiveness in time of each composition of the population [19], here identified by the fraction of Cs. The remarkable agreement between the roots of $G^A(j)$ and the peaks of the stationary distribution gives credit to $G^A(j)$ while emphasizing the fundamental transformation in the evolutionary dynamics of the population introduced by a complex network of interactions. As we show below, the emergence of an unanticipated global (macroscopic) dynamics from a distinct agent (microscopic) dynamics pervades throughout evolutionary dynamical processes in structured populations.

The co-existence point is associated with the internal root of $G^A(j)$, x_R, inexistent in well-mixed populations, and whose location decreases with increasing B. Together with x_R one obtains a coordination root ($x_L \approx 0$) of $G^A(j)$ since, in the absence of cooperative partners, Cs will always be disadvantageous. However, the impact of x_L is minor, as shown in Fig. 1b. In Fig. 2 we track the position of x_R (dots) for two different homogeneous structures along a range of the B values. These are compared with the equilibrium fraction of cooperators (lines), in other words the stationary states. As expected we find a match between the ***AGoS*** prediction and the dynamical outcome, thus providing evidence that the ***AGoS*** remains valid and quantitatively accurate for a broad range of game parameters and different types of homogeneous networks.

Fig. 1a shows that, as we move from a single agent to a population wide perspective, one witnesses the emergence of a new evolutionary dynamics. This new global dynamics has important practical consequences: The fixation time – the time required for Cs to invade the entire population – becomes much larger in homogeneous networks when compared to well-mixed populations (irrespective of the small-world effects associated with random links) as the population spends a large period of time in the vicinity of x_R, mainly when selection is strong (large β).

The analysis in Fig.1 was limited to the time-independent $G^A(j)$ as we averaged over the entire time span of all runs. However, the ***AGoS*** itself evolves in time, giving origin to a time-dependent $G^A(j,t)$. At the beginning of each simulated evolution, Cs and Ds are randomly spread in the network, precluding the occurrence of correlated (assorted) clusters of agentswith the same strategy. Hence, $G^A(j, t = 0) < 0$ in general. As populations evolve, Cs (Ds) breed Cs (Ds) in their neighborhood, promoting the assortment of strategies, with implications both on the fitness of each player and on the shape (and sign) of $G^A(j,t)$. The time-dependent gradients $G^A(j,t)$ for a particular generation t_0 (and corresponding roots) are trivially computed by averaging over the configurations occurring during N previous time-steps (1 generation),

$$G^A(j) = \frac{1}{\Omega\Delta} \sum_{t=t_0-N}^{t_0} \sum_{p=1}^{\Omega} G_p(j,t) \tag{8}$$

In Fig. 3a we plot snapshots of $G^A(j,t)$ for three different times, whereas Fig. 3b portrays the time evolution of the internal roots (x_L and x_R) of $G^A(j,t)$, on which we superimposed two evolutionary runs starting with strategies randomly placed in the population. As $G^A(j, t = 0) < 0$, the fraction of Cs will start decreasing (Fig. 3a). However, with time, strategy assortment leads to the emergence of a co-existence root of $G^A(j,t)$, towards which the fraction of Cs converges. The ensuing coexistence between Cs and Ds, which matches perfectly the shape of $G^A(j,t)$, stems from the evolving self-organization of Cs and Ds in the network, defining a global dynamics which is impossible to predict from the nature of the local (**PD**) interactions.

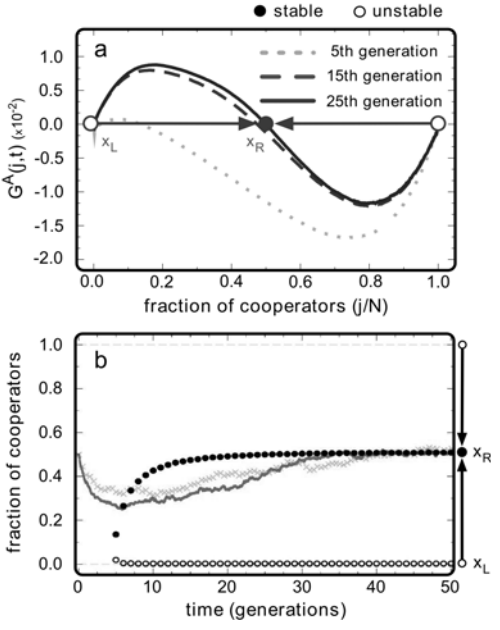


Fig. 3. Time-dependent *AGoS*. (a) We plot $G^A(j,t)$ for three different moments of evolutionary time. Each line provides a snapshot for a given moment, portraying the emergence of a population-wide (time-dependent) co-existence-like dilemma stemming from an agent (time-independent) defection dominant dilemma (**PD**). **(b)** The circles show the position of the different interior roots of $G^A(j,t)$, whereas the solid (dark grey points) line and (light grey crosses) crosses show two independent evolutionary runs starting from 50% of Cs and Ds randomly placed in the networked population. Open (full) circles stand for unstable, x_L (stable, x_R) roots of $G^A(j,t)$ ($B = 1.01$, $N = 10^3$, $k = 4$ and $\beta = 10.0$).

2.4 Results for Heterogeneous Networks

It is now generally accepted that homogeneous networks provide a simplified picture of real interaction networks [3,5,6,11–13]. Most social structures share a marked heterogeneity, where a few nodes exhibit a large number of connections, whereas most nodes comprise just a few. The fingerprint of this heterogeneity is provided by the associated network degree distributions, which exhibit a broad-scale shape, often resembling a power-law [3,5,6,12]. In the following we use $G^A(j,t)$ to show how population heterogeneity shifts the internal roots in Fig. 1 to the right, effectively transforming a co-existence scenario into a coordination one. To this end, we compute $G^A(j,t)$ employing scale-free (**SF**) networks of Barabási and Albert (**BA**) [5], which provide a widely used representation of a heterogeneous structured population [12]. Fig. 4a shows $G^A(j)$ for **BA** networks, whereas the circles in Fig. 4b portray the time evolution of the internal roots of $G^A(j,t)$.

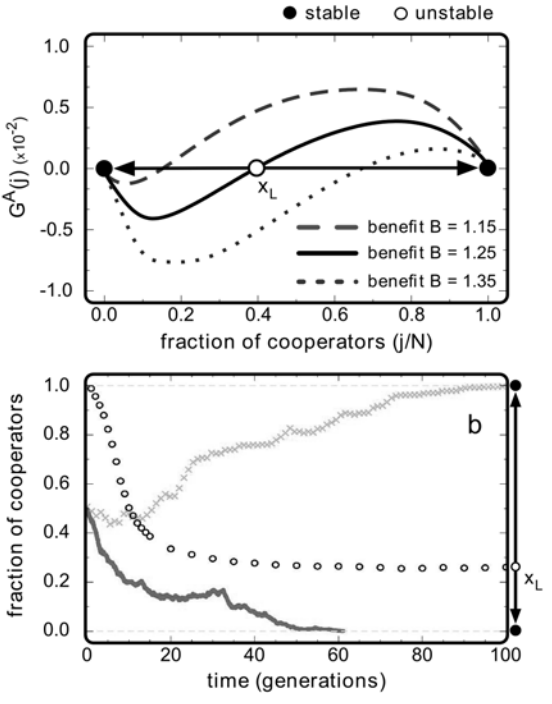


Fig. 4. AGoS on BA networks. (a) Starting from a defection dominance **PD** played at an agent level, a coordination dynamics emerges at a global, population-wide scale, for the three values of B depicted. **(b)** Evolution of the unstable root x_L of $G^A(j,t)$ (open circles), exhibiting the time-dependence of the global dynamics; solid (dark grey dots) line and (light grey crosses) crosses show two independent evolutionary runs starting from 50% of Cs and Ds randomly placed. The ultimate fate of Cs in each run depends on whether the population composition crosses over the time-dependent value x_L of $G^A(j,t)$, thereby overcoming the dynamical coordination barrier during evolution. ($B = 1.25$, $N = 10^3$, $< k >= 4$ and $\beta = 0.1$). BA networks were obtained combining growth and preferential attachment, following the model proposed by Barabási and Albert [5].

Clearly, heterogeneous networks lead to a global dynamics dominated by a coordination threshold x_L. This unstable root of $G^A(j,t)$ represents the critical fraction of Ds above which they are able to assort effectively. Once this happens, they successfully invade highly connected nodes (hubs), rendering cooperation an advantageous strategy, as Cs acquire then a higher probability of being imitated than Ds. The requirement that Cs must first invade hubs before outcompeting Ds (by formation of cooperative star-like clusters [27]), makes invasion harder for isolated Cs. Consequently, the unstable root x_L (located close to $j/N \approx 0$ in homogeneous networks) moves here to higher fractions of Cs. Once this coordination is overcome, Cs benefit from the strong influence of hubs to rapidly spread in the population, eventually leading to fixation. Hence, the stable

internal root x_R which characterizes $G^A(j)$ in homogeneous networks collapses into values close to $j = N$ on **SF** networks, leading to full cooperation. Naturally, the location of x_L is an increasing function of B, as shown in Fig. 4a.

The requirement that Cs occupy the hubs to outcompete Ds also leads to an intricate interplay between the time-dependent decline of x_L (see Fig. 4b) and the pervasiveness of Cs in the population. In Fig. 4b we show, with full lines, two evolutions in **BA** networks (for the same conditions): One ends up in full cooperation whereas the other reaches full defection. In the former, the fraction of Cs decreases in time slower than x_L. Hence, a crossover moment is reached, after which $j/N > x_L$. As a result, the population will subsequently reach full cooperation. In the latter, j/N remains always below x_L and the population evolves towards full defection. Clearly, heterogeneous networks lead to the emergence of a global dynamics with time-dependent coordination barriers and basins of attraction, all of which can be characterized using $G^A(j,t)$.

3 Conclusions

Overall, our study shows that behavioral dynamics in social networks can be understood as if the network structure is absent but agents faced a different dilemma: The structural organization of a population of self-regarding agents circumvent the Nash-equilibrium of a cooperation dilemma by creating a new dynamical system globally described by two internal fixed points, x_L (unstable) and x_R (stable). Moreover, such a dynamical system, resulting from agents interacting via a two-person game, cannot be mapped onto a two-person evolutionary game in a well-mixed population. On the contrary, such dynamics resembles that from, e.g., N-person dilemmas [20] in the presence of coordination thresholds [28, 34]. Hence, the global dynamics of a 2-person dilemma in structured populations resembles a time-dependent N-person dilemma, in which the coordination or co-existence features emerge from the population structure itself. In this sense, different network topologies emphasize differently this co-existence/coordination dichotomy. In such a context, the ***AGoS*** proves instrumental in characterizing the emergence of a new population-wide evolutionary dynamics.

In sum it is of our belief that these results, together with the methodology proposed here are of broad interest for areas within the biological and social sciences that extend far beyond the scope of cooperation problems [6, 11–13]. Moreover, we address a core problem common to most complex systems analysis on fields such as biology, social and engineering sciences: describe the link between local and global dynamics in multi-agent systems. From human behaviors, epidemics, collective intelligence or many population-based applications, most can be described as an interaction scheme embedded in a complex network for which a tool such as the ***AGoS*** may help us to anticipate the emergent, population-wide, global dynamics.

Acknowledgements. The authors acknowledge financial support from FCT-Portugal.

References

1. Adar, E., Huberman, B.: Free riding on gnutella. First Monday 5(10-2) (2000)
2. Akyildiz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. IEEE Communications Magazine 40(8), 102–114 (2002)
3. Amaral, L.A., Scala, A., Barthelemy, M., Stanley, H.E.: Classes of small-world networks. Proceedings of the National Academy of Sciences 97, 11149–11152 (2000)
4. Axelrod, R.: The Evolution of Cooperation. Penguin Books, Harmondsworth (1989)
5. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science 286(5439), 509–512 (1999)
6. Barrat, A., Barthelemy, M., Vespignani, A.: Dynamical processes in complex networks. Cambridge University Press, Cambridge (2008)
7. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm intelligence: from natural to artificial systems, vol. (1). Oxford University Press, USA (1999)
8. Bonabeau, E., Dorigo, M., Theraulaz, G.: Inspiration for optimization from social insect behaviour. Nature 406(6791), 39–42 (2000)
9. Borgers, T., Sarin, R.: Learning through reinforcement and replicator dynamics. Journal of Economic Theory 77(1), 1–14 (1997)
10. Centola, D.: The spread of behavior in an online social network experiment. Science 329, 1194 (2010)
11. Christakis, N.A., Fowler, J.H.: The collective dynamics of smoking in a large social network. New England Journal of Medicine 358(21), 2249–2258 (2008)
12. Dorogovtsev, S.N.: Lectures on Complex Networks. Oxford University Press, USA (2010)
13. Fowler, J.H., Christakis, N.A.: Cooperative behavior cascades in human social networks. Proceedings of the National Academy of Sciences 107(12), 5334–5338 (2010)
14. Gómez-Gardeñes, J., Campillo, M., Floría, L.M., Moreno, Y.: Dynamical organization of cooperation in complex topologies. Physical Review Letters 98(10), 108103 (2007)
15. Granovetter, M.: The strength of weak ties. American Journal of Sociolgy 78, 1360 (1973)
16. Hauert, C.: Effects of space in 2x2 games. International Journal Bifurcation Chaos 12, 1531–1548 (2002)
17. Hofbauer, J., Sigmund, K.: Evolutionary games and population dynamics. Cambridge University Press, Cambridge (1998)
18. Johnson, D., Maltz, D., Broch, J., et al.: Dsr: The dynamic source routing protocol for multi-hop wireless ad hoc networks. Ad Hoc Networking 5, 139–172 (2001)
19. van Kampen, N.: Stochastic processes in physics and chemistry. North-Holland (2007)
20. Kollock, P.: Social dilemmas: The anatomy of cooperation. Annual Review of Sociology 24, 183–214 (1998)
21. Lazer, D., Pentland, A., Adamic, L., Aral, S., Barabasi, A.L., Brewer, D., Christakis, N., Contractor, N., Fowler, J., Gutmann, M., Jebara, T., King, G., Macy, M., Roy, D., Alstyne, M.V.: Computational social science. Science 323(5915), 721–723 (2009)
22. Lloyd, A.L., May, R.M.: How viruses spread among computers and people. Science 292, 1316–1317 (2001)

23. Nakamaru, M., Matsuda, H., Iwasa, Y.: The evolution of cooperation in a lattice-structured population. Journal of Theoretical Biology 184(1), 65–81 (1997)
24. Nowak, M.A., May, R.M.: Evolutionary games and spatial chaos. Nature 359, 826–829 (1992)
25. Ohtsuki, H., Hauert, C., Lieberman, E., Nowak, M.A.: A simple rule for the evolution of cooperation on graphs and social networks. Nature 441(7092), 502–505 (2006)
26. Onnela, J.P., Reed-Tsochas, F.: Spontaneous emergence of social influence in online systems. Proceedings of the National Academy of Sciences 107(43), 18375–18380 (2010)
27. Pacheco, J.M., Pinheiro, F.L., Santos, F.C.: Population structure induces a symmetry breaking favoring the emergence of cooperation. PLoS Computational Biology 5(12), e1000596 (2009)
28. Pacheco, J.M., Santos, F.C., Souza, M.O., Skyrms, B.: Evolutionary dynamics of collective action in n-person stag hunt dilemmas. Proceedings of the Royal Society B 276(1655), 315–321 (2009)
29. Perkins, C., Royer, E.: Ad-hoc on-demand distance vector routing. In: Second IEEE Workshop on Mobile Computing Systems and Applications, WMCSA 1999, pp. 90–100 (1999)
30. Ripeanu, M.: Peer-to-peer architecture case study: Gnutella network. In: Proceedings of First International Conference on Peer-to-Peer Computing, pp. 99–100 (2001)
31. Santos, F.C., Pacheco, J.M.: Scale-free networks provide a unifying framework for the emergence of cooperation. Physical Review Letters 95(9), 98104 (2005)
32. Santos, F.C., Pacheco, J.M., Lenaerts, T.: Evolutionary dynamics of social dilemmas in structured heterogeneous populations. Proceedings of the National Academy of Sciences 103(9), 3490–3494 (2006)
33. Santos, F.C., Rodrigues, J.F., Pacheco, J.M.: Epidemic spreading and cooperation dynamics on homogeneous small-world networks. Physical Review E 72(5), 56128 (2005)
34. Santos, F., Pacheco, J.: Risk of collective failure provides an escape from the tragedy of the commons. Proceedings of the National Academy of Sciences 108(26), 10421 (2011)
35. Sigmund, K.: The Calculus of Selfishness. Princeton Series in Theoretical and Computational Biology. Princeton University Press (2010)
36. Sutton, R., Barto, A.: Reinforcement learning: An introduction, vol. 28. Cambridge University Press, Cambridge (1998)
37. Szabó, G., Fáth, G.: Evolutionary games on graphs. Physics Reports 446(4-6), 97–216 (2007)
38. Taylor, P.D., Day, T., Wild, G.: Evolution of cooperation in a finite homogeneous graph. Nature 447, 469–472 (2007)
39. Traulsen, A., Hauert, C.: Stochastic evolutionary game dynamics, vol. II. Wiley-VCH (2009)
40. Traulsen, A., Pacheco, J.M., Nowak, M.A.: Stochastic dynamics of invasion and fixation. Physical Review E 74(1 Pt 1), 11909 (2006)
41. Van Segbroeck, S., De Jong, S., Nowé, A., Santos, F., Lenaerts, T.: Learning to coordinate in complex networks. Adaptive Behavior 18(5), 416 (2010)
42. Watts, D.J.: A twenty-first century science. Nature 445(7127), 489 (2007)
43. Wooldridge, M., Jennings, N.: Intelligent agents: Theory and practice. Knowledge Engineering Review 10(2), 115–152 (1995)

GeNet: A Graph-Based Genetic Programming Framework for the Reverse Engineering of Gene Regulatory Networks

Leonardo Vanneschi[2,1], Matteo Mondini[1],
Martino Bertoni[1], Alberto Ronchi[1], and Mattia Stefano[1]

[1] D.I.S.Co., University of Milano-Bicocca, 20126, Milan, Italy
[2] ISEGI, Universidade Nova de Lisboa, 1070-312, Lisboa, Portugal

Abstract. A standard tree-based genetic programming system, called GRNGen, for the reverse engineering of gene regulatory networks starting from time series datasets, was proposed in EvoBIO 2011. Despite the interesting results obtained on the simple IRMA network, GRNGen has some important limitations. For instance, in order to reconstruct a network with GRNGen, one single regression problem has to be solved by GP for each gene. This entails a clear limitation on the size of the networks that it can reconstruct, and this limitation is crucial, given that real genetic networks generally contain large numbers of genes. In this paper we present a new system, called GeNet, which aims at overcoming the main limitations of GRNGen, by directly evolving entire networks using graph-based genetic programming. We show that GeNet finds results that are comparable, and in some cases even better, than GRNGen on the small IRMA network, but, even more importantly (and contrarily to GRNGen), it can be applied also to larger networks. Last but not least, we show that the time series datasets found in literature do not contain a sufficient amount of information to describe the IRMA network in detail.

1 Introduction

Biological systems are very complex and it is nowadays recognized that, in order to understand their functioning, we have to analyze the interactions among their components at several different levels [14]. The emerging field of systems biology [8] is aimed at a formal understanding of biological processes via the development of quantitative mathematical models of these interactions. Typically, these models describe biological systems as networks, where regulatory interactions among genes are explicitly represented (gene regulatory networks, or GRNs).

Numerous formalisms to model GRNs have been defined so far (for instance boolean random networks [9] or systems of ordinary differential equations [17,16]). Typically, the data consist of measurements at steady state after multiple perturbations (like gene overexpression, knockdown or drug treatment) or, as considered in the present work, at multiple instants after one or more perturbations (i.e. time series data). Many reverse engineering approaches have been proposed to date, and their assessment and evaluation is of critical importance [15]. In 2011, a system for the GRNs reverse engineering

M. Giacobini, L. Vanneschi, and W.S. Bush (Eds.): EvoBIO 2012, LNCS 7246, pp. 97–109, 2012.

using time series data, based on Genetic Programming (GP) [11,13], called GRNGen (which stands for Gene Regulatory Network Generator) and inspired by other GP based GRN reverse engineering methods defined so far (like [1]), was proposed [5]. The performances of this system were tested on an extremely simple synthetic GRN called IRMA, composed by only five genes and first introduced in [3] (Section 3 contains a brief introduction of this network). Despite its simplicity and the interesting results shown on IRMA, GRNGen presents some important limitations, the most serious one being the fact that a different regression problem has to be solved by GP for each gene composing the GRN to be reconstructed. This fact, by admission of the authors of [5] themselves, is a serious limitation to the scalability of the proposed method, making it practically impossible to use for true GRN containing large numbers of genes. The aim of this paper is presenting a new GP environment that overcomes the major limitations of GRNGen. We call this system *GeNet*. Contrarily to GRNGen, that was based on standard tree-based GP [11], evolving individuals (i.e. potential solutions contained into the population) in GeNet are directly GRNs models, represented as graphs; thus GeNet is a Graph-Based GP system [12,13].

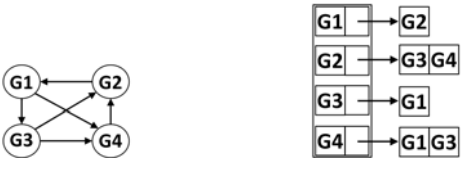
This paper is structured as follows: in Section 2, we present GeNet; in particular, we explain how GRNs can be represented as evolving individuals in a graph-based GP system, the way in which their fitness is calculated, and how we have redefined crossover and mutation. In Section 3 we discuss the case studies used to experimentally validate GeNet: the IRMA network and a set of artificially generated GRNs of several sizes. Section 4 describes our experimental study: on the IRMA network the performances of GeNet are compared with the results reported in [5], including not only the performances of GRNGen, but also the ones of other state of the art GRN reverse engineering methods. On the other hand, the results returned by GeNet on the artificially generated networks are used to discuss GeNet scalability. Finally, Section 5 concludes the paper and discusses ideas for future research.

2 GeNet

In this section we present the new GP system to evolve GRNs; in particular, we define the representation of GRNs that we have adopted, the fitness function and how we have redefined crossover and mutation.

Representation of the Individuals as GRNs. We model a GRN as a directed, colored and weighted graph, composed by a set V of vertices and a set E of edges, $N = \langle V, E \rangle$, where V contains exactly one vertex for each gene of the network and E is a subset of $V \times V$ in which every edge represents a typed (colored) connection between two vertices. Given two vertices $v_1, v_2 \in V$, there is a (directed) edge from v_1 to v_2 (and we write $v_1 \xrightarrow{e} v_2$) if and only if the expression value of the gene represented by v_1 has an influence on the expression value of the gene represented by v_2. The color of the edges represents the type of influence; in our simple model it can be either "+" or "-". Moreover, if we want the model to be as realistic as possible, we cannot assume that every gene expression affects the network in the same proportion, due for instance to different speed and rate of expression, so we add a weight to every edge. In a $v_1 \xrightarrow{e} v_2$ connection the weight indicates the strength of the influence of v_1 on v_2. For each edge

e, we can express the color and the weight of e as functions: $col(e) : E \rightarrow \{+, -\}$ and $w(e) : E \rightarrow \mathbb{R}$. Furthermore, in the cell there might be some genetic product of an inactive gene, called basal production; to model this we added a node constant, that simulates the basal expression value of the gene. Given its nature, it can only assume positive values. Moreover, we also wanted to simulate a persistence of products expressed by a gene in successive timesteps, so we added a persistence constant (it can be imagined as the opposite of a decay value). This constant also allows us to model auto-allostheric promoters (i.e. genes that affect directly their own production by an enzyme that uses positive feedback as an activation method). Thus, every vertex v has two properties: the basal constant and the persistence constant ($const(v) : V \rightarrow \mathbb{R}$ and $pers(v) : V \rightarrow [0, 1] \subseteq \mathbb{R}$). In Figure 1 we report the traditional graph-representation of an extremely simple hypothetical GRN composed by four genes (part (a)) and the corresponding representation that would be used by the GeNet implementation (part (b)). For each gene, we keep a linear structure containing genes from incoming connections (i.e. the other genes of the network that influence it). Given that the evolving individuals in GeNet are GRNs, sometimes in this paper we will use the terminology individuals/networks to address them.



(a) (b)

Fig. 1. Part (a): the graph representation of a very simple hypothetical GRN. Part (b): how this GRN would be represented in the GeNet implementation.

Simulation of the Dynamics of a GRN in GeNet. In order to simulate the dynamics of a GRN (simulation that will be used to calculate the fitness of individuals/networks), we need a function that calculates the new values of expression of the genes using the old ones and the influence information stored in the network. For simplicity, we call this function *influence function* from now on. For each gene, the influence function is calculated using the values of the constants of the gene and the weights of the incoming edges along with the expression values (at the previous timestep) of nodes linked to them. For example, a possible value of the influence function for the gene $G4$ in Figure 1 could be: $G4_t = +(0.4 * G3_{t-1}) - (0.8 * G1_{t-1}) + (0.2 * G4_{t-1}) + 3$ (where we indicate by G_t the expression value of gene G at time t). This means that the expression value of $G4$ at a given time t depends on the expression values of $G1$ (inversely) and $G3$ (directly) at time $t - 1$, and also by the value of $G4$ itself at time $t - 1$. For each influencing gene we have a sign and a multiplicative factor; the sign is determined by the type of the connection and the multiplicative factor is expressed by the weight of the connection, except in the case of $G4$ itself, where the multiplicative factor quantifies the persistence of gene $G4$. Finally, the basal constant of gene $G4$ is added. More formally, the influence function of a gene represented by a vertex $\bar{v}$ in the graph is defined as:

$$Infl(\bar{v},t) = \left(\sum_{\substack{v \in V_{IN} \\ e \in E : v \xrightarrow{e} \bar{v}}} col(e) * w(e) * v_{t-1} \right) + pers(\bar{v}) * \bar{v}_{t-1} + const(\bar{v}) \quad (1)$$

where: V_{IN} is the set of vertices $\{v \mid \exists e \in E : v \xrightarrow{e} \bar{v}\}$; for each edge e, $col(e)$ is its color and $w(e)$ is its weight and for each vertex v, v_t is the expression value of the gene represented by v at time t, $pers(v)$ is the persistence value of v and $const(v)$ is the basal constant value of v, as defined previously. We are aware that activation and inhibition are too complex processes to be represented only by plus/minus sign and we point out that there is not a close direct relation between the sign that a particular gene has in the influence function of another gene (i.e the $col(e)$ value of the edge e connecting them) and the role of the former towards the latter in the real net (i.e. if the former is an inhibitor or an activator of the latter).

Fitness Calculation. Given a target time series dataset, we take the expression values of all the genes at the first time step, and we use them as the starting point to reconstruct the dataset itself, by simulating the dynamics of the network. Successively, in order to obtain the expression value of a gene represented by a vertex $\bar{v}$ at time t, we simply calculate $Infl(\bar{v},t)$ applying equation (1). If we iterate this process for a number of instants equal to the number of time steps of the target dataset, we reconstruct a new dataset of the same dimensions as the target one. At this point, as fitness, we simply use the root mean squared error (RMSE) between the target dataset and the calculated one.

Initialization of the Population. It is nowadays an accepted fact that GRNs generally have a *scale-free* topology, i.e. they are characterized by a large number of nodes poorly connected, and a few nodes richly connected (see for instance [2]). For this reason, we have decided to initialize the population by directly constructing individuals/networks with a scale-free topology. Many algorithms exist in literature to accomplish this task. To initialize GeNet populations we use one of the most simple and well-known one, called *incremental growth with preferential attachment*, that can be found, for instance, in [2]. In synthesis, this algorithm works as follows: to create a random network with a scale-free topology, we start with an empty network. We first insert a vertex in the network, then we insert a second vertex and we link it to the first one. Successively, we define a value, called *likelihood*, associated to each one of the vertices added to the network so far, and proportional to the number of vertices linked to it. So, for example, in the initial situation in which we have added only two nodes (with a link between each other), these nodes will have a likelihood of $1/2$ each. The next step consists in adding a third vertex to the network and link it to one of the already existing vertices, chosen with a probability that, for each vertex, is equal to its likelihood. Successively, we update the likelihoods of all the vertices added to the network so far and we iterate the algorithm until all the vertices that are supposed to take part in the network have been added. The fact that the likelihood is proportional to the connectivity ensures that, at each step, it will be more likely to create an edge connecting a new vertex to an

existing vertex that is already more connected than others, thus generating a scale-free network. At the end of this iterative process, we add edges to the network until a given predefined level of connectivity is reached. We link this new edges to nodes chosen randomly again with probability equal to their likelihood. The algorithm that allows us to initialize the GeNet population is a simple iteration of N independent generations of as many scale-free networks (where N is the predefined population size), where, for each individual/network to be created we generate the desired connectivity at random from a gaussian distribution, whose values of the mean and standard deviation are parameters of the algorithm.

Crossover. Before describing the crossover operator that we have defined for GeNet, it is suitable to point out that, by construction, the initialization algorithm described above creates an initial population where each individual/network has exactly the same vertices (where each vertex represents exactly one gene). This fact will not be changed neither by crossover, nor by mutation; in other words, all the individuals/networks in the population will always have the same vertices (one for each gene). What changes from one individual to the other are the connections between the vertices and the constant values associated to each vertex and to each edge (color and weight of the edges and basal constant and persistence value of the vertices). This is a characteristic that distinguishes GeNet from traditional graph-based GP, as defined for instance in [12,13], where the vertices of a graph (as the nodes of a tree in standard tree-based GP [11,13]) can change from an individual to the other. We believe that this characteristic of GeNet is reasonable, given that GeNet has to reconstruct GRNs from time series datasets, where the number of genes is known and fixed. The crossover operator that we defined takes as input two parental individuals/networks and creates two offsprings. Given two individuals/networks i and j (that represent the parents), crossover works by considering all the pairs of vertices representing the same gene in i and j. In particular, we consider the incoming edges in i and in j for these vertices and: (1) the edges that are in common to both the parents are passed-by to both offsprings; (2) the remaining edges are distributed randomly between the two offsprings (i.e. for each one of these edges we randomly choose with uniform probability the offsprings in which to insert it).

Mutation. We defined five types of mutation for GeNet: (1) the add/removal of an edge; (2) the change of color of an already existing edge; (3) the variation of the basal constant associated to a vertex; (4) the variation of the persistence associated to a vertex; (5) the variation of the weight of an edge. The first two operators are clearly discrete mutations, while the last three ones operate on continuous values, given that they mutate values that are real numbers. We have chosen to implement the latter ones as Gaussian mutations centered in the current value to be mutated; in this way we intend to define mutation operators that are as less ”destructive” as possible.

3 Case Studies

We present the performances of GeNet on two different test cases: the first one is composed by the two datasets provided in [3] for the well-known IRMA network (following [3], these two datasets will be called *switch-on* and *switch-off*). This first test case

allows us to compare the results returned by GeNet with the reverse engineering methods presented in [5] and [3]. The second test case consists in a set of hand-generated networks and the relative time series. The second test case is, as far as we know, new, and so no comparison with computational methods from the literature will be done on it: it is defined just to show some characteristics of scalability of GeNet.

IRMA: Switch-on and Switch-off Datasets. IRMA [3] is a simple synthetic network that contains five genes in the yeast *Saccharomyces cerevisiae*, regulating each other. A description of these genes is beyond the objectives of this paper. The interested reader is referred to [3]. The IRMA network was defined with the explicit goal of enabling easy in vivo reverse engineering and modelling assessment. In IRMA, each gene controls the transcription of at least one other gene within the network, and so the network is connected (see Figure 2(a) for a graph representation of IRMA). In addition, the network can be "switched" on or off by culturing cells in galactose or in glucose, respectively. These actions allow us to set the network in the two possible states that it can assume and allowed the authors of [3] to obtain two different time series datasets, *switch-on* and *switch-off*, that will be also used here. The authors of [3] assert that IRMA can easily be grown and manipulated and it includes a variety of regulatory interactions, thus capturing the behaviour of larger eukaryotic GRNs on a smaller scale. The usefulness of IRMA as a simplified biological model to benchmark both modelling and reverse engineering approaches was tested in [3]. In particular, three well-established reverse engineering approaches were tested on IRMA time series datasets: BANJO (Bayesian network) [18], NIR and TSNI (ordinary differential equations) [7,6]. Results obtained by these approaches, as well as GRNGen [5], are used here for comparison with GeNet.

Artificial Networks. To evaluate the ability of GeNet to work also on large networks, we generated artificial (or "artifact") target networks composed by different numbers of genes. Successively, each one of these networks has been used to generate time series (by simulating the dynamics of the network) of a given number of steps, which we used as the input dataset for GeNet. The procedure we used to generate an artificial network takes as input the number of genes and the desired average connectivity, and it works as follows: for each gene, the constant and persistence values are generated randomly with uniform distribution in the range $[0, 1]$; then, edges are added randomly until the average connectivity is reached. The random generation of an edge simply works by linking together two randomly chosen (with uniform distribution) nodes that are not yet connected. The color of the edge is chosen uniformly at random in the set $\{+, -\}$ (by simulating a coin tossing) and the weight is chosen uniformly at random in the range $[0, 1]$. We generated networks of 5, 25 and 100 genes and for each one of these networks we generated time series datasets composed by 20, 50, 200 and 500 points, obtaining 12 different datasets.

4 Results

We have performed 500 independent runs for the switch-on and switch-off datasets and 50 for the artificial datasets. In each run, we have used the following configuration (obtained after a set of preliminary tests performed for parameter tuning) for GeNet: a

population of 1000 individuals evolved for 100 generations, using tournament selection with a tournament size equal to 30. Elitism was used, copying into the new population the two best individuals at each generation. The used crossover rate was 1, while for each one of the different mutation operators, we used a rate equal to 0.01 (for the add/remove of an edge, when the operator had to be performed, we flipped a coin to determine if an edge had to be added or removed). We obtained best results by restricting the nodes persistence, basal constants and the weights of edges to the range $[0, 1]$. Finally, the artificial networks have been built using an average connectivity equal to 1.6, which is exactly the same as in the IRMA network.

Results on the Switch-on and Switch-off Datasets. As the authors of [5] did for the studied methods, we have considered the networks obtained by GeNet, and for each of them we have studied the following performance measures: (1) Positive Predictive Value (PPV) = TP / (TP + FP) and (2) Sensitivity (Se) = TP / (TP + FN); where TP (True Positives) is the number of links that are both in the reconstructed network and in the true one, FP (False Positives) is the number of links that are in the reconstructed network, but not in the true one, and FN (False Negatives) is the number of links that are not in the reconstructed network, but are contained in the true one. For GeNet (as the authors of [5] did for GRNGen) we calculated PPV and Se on the best individual (i.e. the one with the best fitness in the population) of every run at the last generation, and of all these PPV and Se values, we report the best, the average, the median and the standard deviation. As in [5], we consider PPV and Se on the directed graph, ignoring colors. Table 1 (respectively Table 2) reports the results for the switch-off (respectively switch-on) datasets. Both Tables 1 and 2 contain the results returned by BANJO, NIR/TSNI and GRNGen in the upper part and the results returned by GeNet in the lower part. The results returned by BANJO, NIR/TSNI and GRNGen are exactly the same as reported in [5]. We also point out that (as explained in [5]) BANJO, NIR and TSNI are deterministic methods, and thus we report the results obtained in one execution, instead of reporting the best, average, median and standard deviations of the results obtained over several executions. From Table 1 we can see that, on the switch-off dataset, GeNet outperforms all the other methods in terms of Se (we remark that in the performed runs, we have been able to obtain several networks that have an ideal Se, i.e. Se $= 1$). In particular, GeNet has better values of the best, average and median Se than GRNGen.

Table 1. PPV and Se values returned by the considered methods on the *switch off* data. Upper part: results of BANJO, NIR and TSNI calculated on the networks reported in [3] and results of GRNGen as reported in [5]. Lower part: results obtained by GeNet.

	BANJO	**NIR & TSNI**	**GRNGen (best)**	**GRNGen (median)**	**GRNGen (avg.)**	**GRNGen (std.dev.)**
PPV	0.60	0.60	0.80	0.66	0.66	0.097
Se	0.42	0.42	0.75	0.62	0.58	0.081

	GeNet (best)	**GeNet (avg.)**	**GeNet (median)**	**GeNet (std.dev.)**
PPV	0.62	0.41	0.42	0.07
Se	1.00	0.70	0.75	0.14

Table 2. PPV and Se values returned by the considered methods on the *switch on* data. Upper part: results of BANJO, NIR and TSNI calculated on the networks reported in [3] and results of GRNGen as reported in [5]. Lower part: results obtained by GeNet.

	BANJO	**NIR & TSNI**	**GRNGen (best)**	**GRNGen (median)**	**GRNGen (avg.)**	**GRNGen (std.dev.)**
PPV	0.33	0.75	0.80	0.71	0.68	0.10
Se	0.25	0.42	0.75	0.56	0.57	0.084

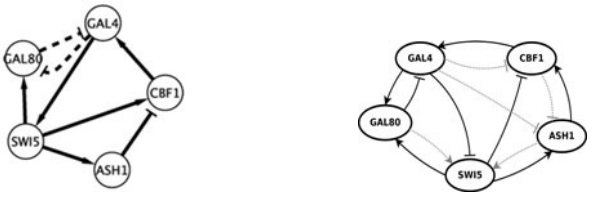
	GeNet (best)	**GeNet (avg.)**	**GeNet (median)**	**GeNet (std.dev.)**
PPV	0.60	0.36	0.36	0.08
Se	1.0	0.58	0.63	0.17

On the other hand, GeNet is outperformed by GRNGen in terms of best, average and median PPV. Similar qualitative considerations hold for the switch-on dataset (results shown in Table 2): GeNet outperforms GRNGen in terms of Se and is outperformed by it in terms of PPV. But, as already pointed out above, the competitive value of GeNet is not in the single results found on the switch-on and switch-off datasets, but in its ability to scale on larger datasets. This is empirically demonstrated in the continuation of this paper. But before considering larger datasets, we feel that another important consideration must be done concerning the results presented so far: in the GeNet experiments, the fitness values[1] of the individuals with the best PPV and Se (reported so far in the tables) are *much* worse than the fitness values of the best individuals that we have obtained (i.e. of the individuals with the best fitness values). Indeed, over the 500 performed runs, the individual with the best fitness *never* corresponded neither with the individual with the best PPV, nor with the individual with the best Se. Even more seriously: the individuals with the best PPV and Se are frequently among the individuals with the *worst* fitness in the population. This discrepancy that we have found between RMSE on data and structural similarity to the IRMA network (quantified by PPV and Se) casts a shadow on the accuracy with which the switch-on and the switch-off datasets describes IRMA. In particular, we believe that single datasets with only 16 (switch-on) and 21 time steps (switch-off) do not contain enough information to describe IRMA by themselves. For the sake of completeness, we also report here the results concerning the fitness values obtained by GeNet on the switch-on and switch-off datasets: for each one of the 500 performed runs, we consider the fitness value of the individual with the best fitness in the population at the last studied generation (these are exactly the same 500 individuals that have been used to report the PPV and Se results in Tables 1 and 2). Of all these 500 fitness values we report the best, average, median and standard deviation in Table 3. Figure 2 reports one of the individuals/networks found by GeNet, that has $PPV = 0.6154$ and $Se = 1.0$.

[1] We remind that the fitness of an individual in GeNet is always calculated as the RMSE between the target time series dataset and the one reconstructed by the individual itself. So, in principle, it has no relationship with the PPV and Se of the network. Furthermore, we also point out that the PPV and Se themselves could *not* have been used as fitness values, because, in order to calculate them, the target network must be known, while reverse engineering methods must work using only the information contained in the time series datasets.

Table 3. Best, average, median and standard deviation of the best fitness obtained in each of the 500 GeNet runs that we have performed on the switch-on and switch-off datasets

	best fitness	**average fitness**	**median fitness**	**fitness std.dev.**
switch-off	0.029	0.045	0.043	0.008
switch-on	0.106	0.129	0.128	0.009



(a) (b)

Fig. 2. Part (a): The topology of the regulatory interactions among genes in IRMA. Dashed lines represent protein-protein interactions. Directed edges with an arrow end represent activation, whereas a dash end represents inhibition. Figure taken from [3]. Part(b): Graphical representation of an individual found by GeNet.

Results on the Artificial Networks. All the results that we have obtained on the artificial networks are reported in Table 4, where a label like "Art$n$" indicates an artificial dataset generated by a network of n genes. Art5, Art25 and Art100 have been simulated to generate datasets of 20, 50, 100 and 500 time steps. We executed 50 indipendent runs and the results are shown in the different tabular forms of Table 4. GeNet is clearly scalable with the number of considered time steps, in the sense that its ability to reconstruct the target network does not change much when passing from datasets with a small (i.e. equal to 20) to a large (i.e. equal to 500) number of time steps. As expected, the performance of GeNet in terms of PPV, Se and fitness deteriorates as the number of genes increases, but one encouraging result emerges from Table 4: the ratio between PPV and number of genes and between Se and number of genes is practically constant for all the studied experiments. It is also worth noticing that obtaining the same results with GRNGen would have been practically impossible (for instance for the networks with 100 genes it would have required the resolution of 100 different regression problems with GP). Another interesting result is the execution time that GeNet has employed to complete the 50 independent runs on the different datasets. These results are shown in Table 5.

On the Consistency between IRMA Topology and Data. The aim of this section is to further investigate the relationship between the fitness of the GeNet individuals/networks and the structural similarity with IRMA. In particular, we want to verify if an individual *identical to* (or at least structurally "*very similar to*") the IRMA network exists in the search space. To reach this goal, we used Particle Swarm Optimization (PSO) [10,4] as a method to optimize continuous (in our case real-valued) parameters. In fact, from [3] we know exactly the topology of IRMA, but we lack knowledge on the values of the parameters (nodes constants, nodes persistences and edges weights) that

Table 4. Results on the artificial datasets. Art$n$ indicates an artificial dataset generated by a network of n genes. Each network has been run for 20, 50, 200 and 500 steps.

20 time steps:

	Art5 (best)	**Art5 (avg.)**	**Art5 (med.)**	**Art5 (s.d.)**	**Art25 (best)**	**Art25 (avg.)**	**Art25 (med.)**	**Art25 (s.d.)**	**Art100 (best)**	**Art100 (avg.)**	**Art100 (med.)**	**Art100 (s.d.)**
PPV	0.636	0.481	0.462	0.077	0.175	0.073	0.070	0.041	0.039	0.014	0.012	0.009
Se	1.000	0.713	0.750	0.134	0.175	0.080	0.075	0.044	0.044	0.016	0.013	0.010
Fitness	0.019	0.040	0.039	0.013	0.055	0.069	0.066	0.010	0.177	0.242	0.238	0.041

50 time steps:

	Art5 (best)	**Art5 (avg.)**	**Art5 (med.)**	**Art5 (s.d.)**	**Art25 (best)**	**Art25 (avg.)**	**Art25 (med.)**	**Art25 (s.d.)**	**Art100 (best)**	**Art100 (avg.)**	**Art100 (med.)**	**Art100 (s.d.)**
PPV	0.700	0.437	0.429	0.098	0.180	0.063	0.065	0.035	0.036	0.018	0.020	0.010
Se	0.875	0.608	0.625	0.152	0.225	0.073	0.075	0.042	0.038	0.019	0.019	0.010
Fitness	0.008	0.027	0.026	0.011	0.028	0.039	0.037	0.009	0.185	0.249	0.242	0.037

200 time steps:

	Art5 (best)	**Art5 (avg.)**	**Art5 (med.)**	**Art5 (s.d.)**	**Art25 (best)**	**Art25 (avg.)**	**Art25 (med.)**	**Art25 (s.d.)**	**Art100 (best)**	**Art100 (avg.)**	**Art100 (med.)**	**Art100 (s.d.)**
PPV	0.667	0.418	0.444	0.124	0.184	0.066	0.059	0.041	0.034	0.014	0.014	0.009
Se	0.875	0.440	0.500	0.160	0.225	0.075	0.075	0.044	0.044	0.016	0.019	0.010
Fitness	0.037	0.037	0.037	0.000	0.030	0.49	0.045	0.017	0.167	0.253	0.250	0.038

500 time steps:

	Art5 (best)	**Art5 (avg.)**	**Art5 (med.)**	**Art5 (s.d.)**	**Art25 (best)**	**Art25 (avg.)**	**Art25 (med.)**	**Art25 (s.d.)**	**Art100 (best)**	**Art100 (avg.)**	**Art100 (med.)**	**Art100 (s.d.)**
PPV	0.600	0.375	0.357	0.137	0.076	0.069	0.065	0.034	0.037	0.017	0.018	0.008
Se	0.875	0.443	0.500	0.202	0.093	0.075	0.075	0.045	0.038	0.019	0.019	0.008
Fitness	0.022	0.026	0.026	0.000	0.022	0.052	0.044	0.024	0.152	0.240	0.232	0.042

Table 5. The completion time that has been necessary for GeNet to execute 50 independent runs on the different datasets. The upper column represents the datasets with the following notation: x-y stands for a dataset of x genes and y timesteps. The lower column reports the execution times in the format hh:mm'ss".

5-20	**5-50**	**5-200**	**5-500**	**25-20**	**25-50**	**25-200**	**25-500**	**100-20**	**100-50**	**100-200**	**100-500**
6'31"	8'53"	16'09"	35'39"	16'02"	25'29"	1:04'13"	2:22'25"	1:10'32"	1:53'18"	5:09'42"	8:47'11"

we must use to exactly represent IRMA in our model. Thus, we have performed 50 runs of PSO using the following configuration: (1) Given that the graph that represents the IRMA network contains 5 vertices (genes) and 8 edges, a particle is a vector composed by 18 real numbers: two real numbers for each vertex (one for the node constant and one for the node persistence) and one real number for each edge (its weight). (2) The fitness of each particle is the RMSE between the target dataset (in these experiments we have chosen the switch-off dataset, given that it is the one for which we have found the best fitness results so far) and the dataset obtained by simulating a network with exactly the same topology as IRMA, but using the constant values represented by the particle

itself. We point out that the fitness used by this PSO system is exactly the same as the one used by GeNet. The only difference is that the search space of the PSO system is restricted to networks that have the same topology as IRMA (what changes is just the values of the constants). Furthermore, even though this may be an obvious consideration, it is also worth pointing out that this technique cannot be used as a feasible reverse engineering method to derive GRNs, because it is based on the previous knowledge of the topology of the target network (which is exactly what reverse engineering methods are supposed to discover). So, here we use this PSO system only to verify that, if we force a network to have exactly the same topology as IRMA, we can find values of the constants that allow us to perfectly match the time series data. In other words, we use this PSO system to verify if IRMA can suitably be explained by the data.

The PSO has been executed with the following parameters setting: inertia = 0.01; cognitive constant C1 = 1.9; social constant C2 = 1.9; number of particles in the swarm = 500; number of iterations = 1000. Furthermore, the possible weights of the edges have been restricted to the range $[0, 3]$ and the nodes constants to the range $[0, 1]$. An experiment made by 50 PSO runs returned the following results: the best particle had a fitness value equal to 0.068; the average fitness (calculated over the best particle found in each of the 50 runs) was equal to 0.151 with a standard deviation of 0.043. Surprisingly, these results are *worse* than the ones returned by GeNet itself (reported in the upper row of Table 3), while we were expecting that a network with the same topology as IRMA should produce much better results. To the best of our consideration, this result can have only one reasonable explanation: there are *a lot* of different networks, even very different from each other, that can fit the data well. Even more importantly, many possible networks with a *completely different topology* from IRMA explains the data much better than IRMA itself as already indicated also by the GeNet results. This is a further, and we believe quite clear, indication of the fact that the switch-off dataset is not informative enough to describe the IRMA network[2].

5 Conclusions and Future Work

One of the main motivations that drove us to define the GeNet system was that, differently from GRNGen introduced in [5], it allows us to evolve entire networks. In fact, in order to reconstruct a network of N genes with GRNGen, one must solve N regression problems with GP and then combine together the obtained results, and this fact clearly limits the dimensions of the networks that can reasonably be reconstructed by GRNGen. The results presented in this paper confirm that GeNet permits us to overcome this limitation, by allowing us to reconstruct networks of several different dimensions in a quite reliable way. Furthermore, this paper also shows that, even on the small network used to present GRNGen in [5] (i.e. the well known IRMA network introduced in [3]), GeNet obtains comparable (and in some case even better) results than GRNGen. Last but not least, this work has allowed us to discover interesting facts about the IRMA network. In particular, we have clearly shown that the switch-on and switch-off datasets reported in [3] do not contain a sufficient amount of information to describe IRMA in detail.

[2] Exactly the same qualitative conclusions can be drawn for the switch-on dataset, but we do not report the results for lack of space caused by the strict page limit imposed to this publication.

Our current research activity consists in testing the GeNet system on real (and very large) GRNs. The results that our work will produce will in any case be an interesting contribution, because, given the large number of genes, they simply could not have been obtained using GRNGen. But, of course, we also want those results to have the best possible quality. For this reason, while running the experiments on real networks, we are also still working on some adjustments, improvements and details of the current GeNet version. For instance, we are looking for the most effective parameter setting and we are trying to define more effective genetic operators. We consider this study as a long term and ambitious one, but we believe that this paper, introducing GeNet for the first time, can represent the very first important step in this research track.

Acknowledgments. Leonardo Vanneschi gratefully acknowledges project PTDC/EIACCO/103363/2008 from Fundação para a Ciência e a Tecnologia, Portugal.

References

1. Banzhaf, W.: Artificial regulatory networks and genetic programming. In: Riolo, R.L., Worzel, B. (eds.) GP Theory and Practice, ch. 4, pp. 43–62. Kluwer (2003)
2. Barabasi, A.-L.: Linked: How Everything Is Connected to Everything Else and What It Means for Business, Science, and Everyday Life. Plume Books (April 2003)
3. Cantone, I., Marucci, L., Iorio, F., Ricci, M.A., Belcastro, V., Bansal, M., Santini, S., di Bernardo, M., di Bernardo, D., Cosma, M.P.: A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches. Cell 137(1), 172–181 (2009)
4. Clerc, M. (ed.): Particle Swarm Optimization. ISTE (2006)
5. Farinaccio, A., Vanneschi, L., Provero, P., Mauri, G., Giacobini, M.: A New Evolutionary Gene Regulatory Network Reverse Engineering Tool. In: Giacobini, M. (ed.) EvoBIO 2011. LNCS, vol. 6623, pp. 13–24. Springer, Heidelberg (2011)
6. Gardner, T.S., Bernardo, D.D., Lorenz, D., Collins, J.J.: Inferring genetic networks and identifying compound mode af action via expression profiling. Science 301, 102–105 (2003)
7. Gatta, G.D., Bansal, M., Ambesi-Impiombato, A., Antonini, D., Missero, C., Bernardo, D.D.: Direct targets of the TRP63 transcription factor revealed by a combination of gene expression profiling and reverse engineering. Genome Res. 18, 939–948 (2008)
8. Hayete, J., McMillen, D., Collins, J.J.: Size matters: network inference tackles the genome scale. Mol. Syst. Biol. 3, 77 (2007)
9. Kauffman, S.A.: The Origins of Order. Oxford University Press, New York (1993)
10. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proc. IEEE Int. Conf. on Neural Networks, vol. 4, pp. 1942–1948. IEEE Computer Society (1995)
11. Koza, J.R.: Genetic Programming. The MIT Press, Cambridge (1992)
12. Niehaus, J., Igel, C., Banzhaf, W.: Reducing the number of fitness evaluations in graph genetic programming using a canonical graph indexed database. Evol. Comput. 15, 199–221 (2007)
13. Poli, R., Langdon, W.B., McPhee, N.F.: A field guide to genetic programming (2008), http://lulu.com, http://www.gp-field-guide.org.uk
14. Sprinzak, D., Elowitz, M.B.: Reconstruction of genetic circuits. Nature 438, 443–448 (2005)

15. Stolovitzky, G., Monroe, D., Califano, A.: Dialogue on reverse-engineering assessment and methods: the dream of high-throughput pathway inference. Ann. N Y Acad. Sci. 1115, 1–22 (2007)
16. Szallasi, Z., Stelling, J., Periwal, V.: System modeling in cellular biology: From concepts to nuts and bolts. The MIT Press, Boston (2006)
17. Ventura, B.D., Lemerle, C., Michalodimitrakis, K., Serrano, L.: From in vivo to in silico biology and back. Nature 443, 527–533 (2006)
18. Yu, J., Smith, V.A., Wang, P.P., Hartemink, A.J., Jarvis, E.D.: Advances to bayesian network inference for generating casual networks from observational biological data. Bioinformatics 20, 3594–3603 (2004)

Comparing Multiobjective Artificial Bee Colony Adaptations for Discovering DNA Motifs

David L. González-Álvarez, Miguel A. Vega-Rodríguez,
Juan A. Gómez-Pulido, and Juan M. Sánchez-Pérez

University of Extremadura,
Department of Technologies of Computers and Communications,
ARCO Research Group,
Escuela Politécnica, Campus Universitario s/n, 10003. Cáceres, Spain
{dlga,mavega,jangomez,sanperez}@unex.es

Abstract. Multiobjective optimization is successfully applied in many biological problems. Currently, most biological problems require to optimize more than one single objective at the same time, resulting in Multiobjective Optimization Problems (MOP). In the last years, multiple metaheuristics have been successfully used to solve optimization problems. However, many of them are designed to solve problems with only one objective function. In this work, we study several multiobjective adaptations to solve one of the most important biological problems, the Motif Discovery Problem (MDP). MDP aims to discover novel Transcription Factor Binding Sites (TFBS) in DNA sequences, maximizing three conflicting objectives: motif length, support, and similarity. For this purpose, we have used the Artificial Bee Colony algorithm, a novel Swarm Intelligence algorithm based on the intelligent behavior of honey bees. As we will see, the use of one or another multiobjective adaptation causes significant differences in the results.

Keywords: Artificial Bee Colony, Swarm Intelligence, DNA, motif discovery, multiobjective optimization.

1 Introduction

Optimization is the process to find the best solution to a given problem, satisfying a set of constrains. When the problem defines only a single objective function, the procedure is simple, we have to find the best solution (called *global optimum*), or a good approximation to it. However, when designing optimization models, often we need to optimize more than one objective function at the same time, these problems are known as Multiobjective Optimization Problems (MOP) [1]. In MOPs the objectives are in conflict with each other, i. e., the optimization of one objective causes a worsening in the others. Therefore, we do not have a single optimal solution, but a set of optimal solutions known as *Pareto set*. The solutions of this set are called *nondominated* solutions, and the plot of the objective function values of these nondominated solutions results in

M. Giacobini, L. Vanneschi, and W.S. Bush (Eds.): EvoBIO 2012, LNCS 7246, pp. 110–121, 2012.

the *Pareto front*. Metaheuristics are one of the most widely used techniques to solve MOPs. From the several existing metaheuristics, evolutionary algorithms are the most popular ones [2]. In this work, we tackle a MOP with different multiobjective adaptations of the Artificial Bee Colony (ABC) algorithm [3]. ABC is a novel collective intelligence algorithm based on the foraging behavior of honey bee swarms. It has been successfully applied to solve many optimization problems, and it has never been used to solve this kind of problems. The problem addressed in our study is a multiobjective optimization model of one of the most important biological problems, the Motif Discovery Problem (MDP), applied to the specific task of discovering novel Transcription Factor Binding Sites (TFBS) in DNA sequences [4]. As we will see, the use of a multiobjective adaptation or another causes significant differences in the quality of the results. So, we must choose carefully the multiobjective adaptation used in our metaheuristics when solving a problem. Our main objective is to use standard concepts and functions to adapt the ABC algorithm to the multiobjective context, since currently there is not defined a multiobjective ABC. To provide a possible methodology for choosing the best multiobjective adaptation for ABC and other algorithms, is the main motivation of this paper. We have defined five multiobjective adaptations of ABC. The first one defines a multi-term fitness function which assigns a weight to each objective to obtain a single fitness value. The second multiobjective adaptation uses the ranking and sorting methodology proposed by Fonseca and Fleming in [5]. We have also experimented with a new methodology that uses as a selection criterion the Hypervolume indicator (HV, [6]). With this multiobjective indicator we can calculate the percentage of the problem search space covered by one or more individuals, allowing us to know which solution is able to cover a larger volume. This indicator is also used, in addition to the Coverage Relation (CR, [7]), to compare the results obtained by our ABC multiobjective adaptations. Finally, we have tested two ABC adaptations where we apply the most important functions of two standard multiobjective algorithms: NSGA-II [8] and SPEA2 [9]. It is important to emphasize that we have conducted a more detailed study of the best multiobjective adaptation, comparing its predictions with those discovered by fourteen well-known biological methods such as: AlignACE, MEME, or Weeder.

The rest of the paper is organized as follows. Section 2 presents a brief review of existing works related to the MDP. It also defines the multiobjective formulation of the problem. In Section 3 we include a description of the applied evolutionary algorithms, detailing the operation of each multiobjective adaptation. Section 4 is devoted to the experimentation, conducting a discussion of the results. Finally, we present some conclusions and future lines in Section 5.

2 Motif Discovery Problem: Review and Formulation

In this section we present a brief review of several methods and techniques used for finding motifs. Then, we explain the multiobjective formulation applied to define the MDP, including a small example to facilitate the understanding of the concepts.

2.1 Brief Review of MDP Related Works

Several approaches in the literature organize the motif finding methods into two groups: methods based on probabilistic models, and string-based methods. Probabilistic methods represent the motifs by a position weight matrix and they are usually designed to find longer or more general motifs. The most popular methods are Consensus, MEME, AlignACE, ANN_Spec, Improbizer, MotifSampler, GLAM, or the recently proposed SeSiMCMC. On the other hand, string-based techniques are appropriate for finding totally constrained motifs, some examples are Oligo/Dyad-Analysis, MITRA, YMF, QuickScore, or Weeder. Thanks to the work [10] we can compare the results obtained by all these fourteen biological methods with those obtained by our algorithm.

In the last years, there have appeared many proposals that use evolutionary computation to solve the MDP. Most of these proposals are based on genetic algorithms (GA), some examples are FMGA [11], St-GA [12], and MDGA [13]. Regarding to the not based on genetic algorithm techniques, we highlight the TS-BFO algorithm [14], and DE/EDA [15]. All these methods employ a single objective implementation, and the motif length is given beforehand, assuming only one motif per sequence. Moreover, almost all of these algorithms try to find motifs in all of the given sequences. In [16] and [17] the authors propose a new multi-term fitness function. The objective of this process was to maximize the similarity of the motifs, while avoiding saturation of low complexity solutions. However, the best way to address the problems previously listed is using a multiobjective approach. Kaya [18] proposed a multiobjective GA-based method named MOGAMOD for discovering motifs, defining an effective multiobjective formulation to address the MDP. Our algorithm uses this multiobjective formulation, incorporating some biological constraints which allow us to better adapt it to the real world.

2.2 Multiobjective Formulation of MDP

We define three conflicting objectives to tackle the MDP: motif length, support, and similarity. Given a set of sequences $S = \{S_i | i = 1, 2, ..., D\}$ of nucleotides defined on the alphabet $B = \{A, C, G, T\}$ we define:

- A nucleotide sequence as $S_i = \{S_i^j | j = 1, 2, ..., w_i\}$, where w_i is the sequence width.
- The set of all the subsequences contained in S as $\{s_i^{j_i} | i = 1, 2, ..., D, j_i = 1, 2, ..., w_i - l + 1\}$, where j_i is the binding site of a possible motif instance s_i^j on sequence S_i, and l is the *motif length*.
- The consensus motif as a string abstraction of the motif instances.

Support is calculated by comparing each candidate motif with the consensus motif, only those sequences that achieve a candidate motif of certain quality with respect to the consensus motif (>=50%), will be taken into account when we build the final motif. The final number of sequences used to build the final solution is indicated by the *support*. To obtain the similarity value we also have to define:

- The Position Indicator Matrix (PIM) $A = \{A_i | i = 1, 2, ..., D\}$ of a motif, where $A_i = \{A_i^j | j = 1, 2, ..., w_i\}$ is the indicator row vector with respect to a sequence S_i, and where A_i^j is 1 if the position j in S_i is a binding site, and 0 otherwise.
- The Position Count Matrix (PCM) $N(A)$ with the numbers of different nucleotide bases on each position of the candidate motifs (A) as $N(A) = \{N(A)^1, N(A)^2, ..., N(A)^l\}$, where $N(A)^j = \{N(A)_b^j | b \in B\}$ and $N(A)_b^j = |\{S(A)_i^j | S(A)_i^j = b\}|$.
- The Position Frequency Matrix (PFM) as $\widehat{N} = \frac{N(A)}{|A|}$ generated by the normalization of the dominant nucleotides of each position.

We calculate the final *similarity* value by averaging all the values of each PFM column. As is indicated in the following expression:

$$Similarity(Motif) = \frac{\sum_{i=1}^{l} max_b\{f(b,i)\}}{l} \tag{1}$$

where $f(b,i)$ is the score of nucleotide b in column i in the PFM and $max_b\{f(b,i)\}$ is the value of the dominant nucleotide in column i.

To guide the pattern search to solutions that have biological relevance, we have incorporated several constraints that should be satisfied by each solution:

- The motif length is restricted to the range [7,64].
- We set a minimum support value of 2 for the motifs of the data sets composed by 4 sequences, and of 3 for the other ones.
- We apply the complexity concept [16] expanded with the improvements suggested in [17] as a biological constraint. The complexity of the candidate motifs should be considered in order to avoid low complexity solutions by using the following expression:

$$Complexity = \log_{10} \frac{l!}{\prod n_i!} \tag{2}$$

where l is the motif length, and n_i is the number of nucleotides of type $i \in \{A, C, G, T\}$.

2.3 MDP Example

In Figure 1 we include an MDP example with $motif\ length = 9$. This example clarifies the methodology followed to obtain the values of the three defined objectives. First, we build the consensus motif by using the candidate motifs. Thus, we can check which candidates exceed the threshold value of support, i. e., which candidates to consider and which not. In this example, five candidates exceed this threshold value, then we have $support = 5$. By using these candidate motifs we have also to build the PCM and the PFM with the occurrence rate of each base at each motif position. Finally, to obtain the similarity value, we apply Equation 1 with the values obtained by the dominant nucleotide at each motif position, in this example we have $similarity = 0.911$.

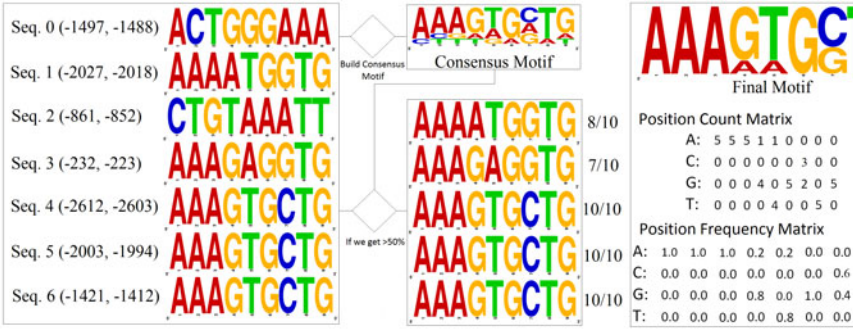


Fig. 1. An artificial motif discovery problem. From left to right it shows the candidate motifs, the consensus motif and the candidates which have surpassed the threshold value of Support, and lastly, the final motif, the position count matrix, and the position frequency matrix.

3 Artificial Bee Colony

In this section we describe the operation of the Artificial Bee Colony (ABC) algorithm. Then, we describe the five multiobjective adaptations presented to solve the MDP.

3.1 Original ABC Algorithm

Artificial Bee Colony (ABC) is a novel Swarm Intelligence based algorithm proposed by Karaboga [3]. It defines an optimization model based on the foraging behavior of honey bees. The model consists of four main components: employed, onlooker, scout foraging bees, and food sources. Employed bees exploit nectar of the food sources, onlooker bees analyze the vicinity of the exploited food sources, and the scout bees provide randomness to the process, exploring more remote areas.

A general outline of ABC is shown below [3]:
1: Initialize randomly the population of solutions x_i (employed bees)
2: Evaluate population
3: Until stopping criterion is not met, perform steps 4-8
4: Produce new solutions for employed bees: $v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj})$, applying a greedy selection process between x_i and v_i
5: Calculate the probability values P_i for the solutions x_i
6: Produce new solutions for onlooker bees, applying the same greedy selection process than in step 4
7: Insert new randomly generated solutions for scout bees, expanding the covered problem search space
8: Rank all solutions to assign the best food sources to the employed bees of the next generation

3.2 Multiobjetive ABC Algorithm

The multiobjective dominance concept [1] indicates that all solutions belonging to the same Pareto front are equally good. If we analyze the outline of ABC, we can detect in which parts there are conflicts for selecting the best solution (or solutions) from a given set of individuals. In steps 4 and 6 we have two possible conflicts when generating and selecting new solutions for employed and onlooker bees. In step 8, we can find another conflict when we have to select the best food sources for the employed bees of the next generation. To know what solutions we have to choose in each case, we must define a multiobjective selection methodology that allow us to know which individual is the best. To address this selection problems, we define five multiobjective adaptations based on several multiobjective techniques and indicators. Now, we describe the operation of each multiobjective adaptation utilized to address these conflicts, including the pseudocode used to tackle the multiobjective sorting function of step 8.

The first adaptation (MOABCv1) addresses a MOP as a single-objective problem. It assigns weights (equal to 1/3, since all three objectives are equally important) to each objective, performing then a normalized sum of their values. Thus, we obtain a single fitness value that allows us to make comparisons without any difficulty. Its multiobjective sorting function is detailed in Algorithm 1. The second adaptation (MOABCv2) uses the rank-based fitness assignment method proposed by Fonseca and Fleming in [5]. Applying this technique, we obtain a single fitness value for each solution, and we can solve the selection conflicts described above. The pseudocode of the sorting function of this second adaptation is shown in Algorithm 2. The third multiobjective ABC adaptation (MOABCv3) bases its behavior on the dominance concept and on an indicator widely used in multiobjective evolutionary computation, the HV indicator [6]. First, we organize the population by fronts, and then, we apply the HV to check what solution covers a larger volume of the problem search space. This version uses the HV to rank the solutions in each Pareto front. By using this new methodology, we solve the problems for choosing the best multiobjective solutions. We show its multiobjective sorting function in Algorithm 3. The following multiobjective adaptation (MOABCv4) is based on the main concepts defined by a standard multiobjective evolutionary algorithm such as NSGA-II [8]. MOABCv4 applies two key functions of NSGA-II: the nondominated sort and the crowding distance calculation. The first one organizes the population by Pareto fronts, as happened in MOABCv3, providing a provisional ranked population. In the second function, we calculate the crowding distance value of each

Algorithm 1. MOABCv1 - *multi-term fitness*

Input: colony C
Output: sorted colony, the first half of bees made up the new population of employed bees
1: */* We obtain the multi-criteria fitness for each colony bee */*
2: **for** i = 1 to ColonySize **do**
3: C[i].MCFitness $\Leftarrow w_1$(C[i].length) + w_2(C[i].support) + w_3(C[i].similarity)
4: **end for**
5: C $\Leftarrow$ sortColonyUsingMCFitness(C)

Algorithm 2. MOABCv2 - *rank-based fitness*

Input: colony C
Output: sorted colony, the first half of bees made up the new population of employed bees
1: */* We sort the colony by using a rank-based fitness assignment */*
2: **for** i = 1 to ColonySize **do**
3: C[i].rank $\Leftarrow 1 + p_i^{(t)}$ //where $p_i^{(t)}$ is the number of solutions that dominates x_i
4: **end for**
5: C $\Leftarrow$ sortColonyUsingRank(C)

Algorithm 3. MOABCv3 - *ranking + hypervolume*

Input: colony C
Output: sorted colony, the first half of bees made up the new population of employed bees
1: */* We sort the colony by using the hypervolume covered by each solution*/*
2: **for** i = 1 to ColonySize **do**
3: C[i].rank $\Leftarrow$ nondominatedRanking(i) //i_{rnk}
4: C[i].hv $\Leftarrow$ calculateSolutionHypervolume(i) //i_{hv}
5: **end for**
6: */* We define an order* $\prec_n$ *as* $i \prec_n j$ *if* $((i_{rnk} < j_{rnk}) or((i_{rnk} = j_{rnk}) and (i_{hv} > j_{hv})))$ **/*
7: C $\Leftarrow$ sortColonyUsingRankingAndHypervolume(C)

Algorithm 4. MOABCv4 - *ranking + crowding distance*

Input: colony C
Output: sorted colony, the first half of bees made up the new population of employed bees
1: */* We sort the colony by using the two key features of the NSGA-II algorithm */*
2: **for** i = 1 to ColonySize **do**
3: C,#paretoFronts $\Leftarrow$ nondominatedSort(C) //i_{rnk}
4: **for** j = 1 to #paretoFronts **do**
5: C $\Leftarrow$ crowdingDistanceCalculation(C,j) //i_{cw}
6: **end for**
7: **end for**
8: */* We define an order* $\prec_n$ *as* $i \prec_n j$ *if* $((i_{rnk} < j_{rnk}) or((i_{rnk} = j_{rnk}) and (i_{cw} > j_{cw})))$ **/*
9: C $\Leftarrow$ sortColonyUsingRankingAndCrowding(C)

Algorithm 5. MOABCv5 - *strength + raw fitness*

Input: colony C
Output: sorted colony, the first half of bees made up the new population of employed bees
1: */* We sort the colony by using the two key features of the SPEA2 algorithm */*
2: **for** i = 1 to ColonySize **do**
3: C[i].strength $\Leftarrow$ calculateBeeStrength(i)
4: **end for**
5: **for** i = 1 to ColonySize **do**
6: C[i].rawFitness $\Leftarrow$ calculateBeeRawFitness(i)
7: **end for**
8: C $\Leftarrow$ sortColonyUsingRawFitness(C)

solution in order to sort the individuals within each Pareto front. The crowding distance concept serves as an estimation of the cuboid perimeter formed by using the nearest neighbors as vertices, and lets us to know which solutions provides greater spreads. The general outline of its multiobjective sorting function is indicated in Algorithm 4. Finally, we propose MOABCv5 that incorporates the main characteristics of other well-known multiobjective evolutionary algorithm, the SPEA2 algorithm [9]. In this algorithm, the fitness assignment strategy (also called *raw fitness*) is divided into two parts: a first, where we determine the number of solutions that dominates each individual (*strength*), and a second, where we consider the number of individuals by which each individual is dominated.

The sum of the individual forces (strengths) that dominate a given solution is its raw fitness. Note that the lower is the force with which an individual is dominated, the better is the solution. In Algorithm 5, we include the sorting function used in MOABCv5.

4 Experimental Results

In this section we explain the methodology followed to configure each algorithm, describing the instances used in our experimentation. We also include the results achieved by the designed adaptations, comparing the obtained results with those obtained by two standard multiobjective algorithms (NSGA-II [8] and SPEA2 [9]), and with those obtained by fourteen well-known biological methods.

In each experiment we have performed 30 independent runs to ensure the statistical significance in the results. We have used the HV [6] as multiobjective metrics, calculating the reference volume from the maximum value of each objective in each instance. To analyze the performance of the algorithms, we have also used the CR [7] indicator, which is useful to know what algorithm gets the best Pareto fronts. All experiments have been performed on a Pentium 4 (2.8 GHz) with 1 GB of RAM by using *gcc* without optimization options. As benchmark, we have used twelve real sequence data sets selected from TRANSFAC database [19]. In Table 1, we describe the properties of each instance. In Table 1, we also include the established runtimes (in seconds). Figure 2 shows the individual representation used in our algorithms. It represents the motif length, and the starting positions of each candidate motif in each sequence. Finally, we have adjusted the parameter values of the algorithms to obtain the best configuration to solve the MDP. The parameter values used are the same as in [20]. Once defined the followed methodology and the parameter settings, we can proceed to compare the results obtained by the multiobjective adaptations. The first comparison uses the HV [6], and it is shown in Table 2. Analyzing the results of Table 2, we notice how the second multiobjective adaptation (MOABCv2, which applies the fitness assignment method proposed by Fonseca and Fleming [5]) achieves the

Table 1. Data set properties

	#Seq.	Size	#Nucl.	Time (s)
dm01g	4	1500	6000	15
dm04g	4	2000	8000	15
dm05g	5	2500	12500	15
hm03r	10	1500	15000	25
hm04m	13	2000	26000	25
hm16g	7	3000	21000	15
mus02r	9	1000	9000	15
mus03g	4	1500	6000	15
mus07g	12	500	6000	25
yst03m	8	500	4000	15
yst04r	7	1000	7000	15
yst08r	11	1000	11000	25

	Seq. 0	**Seq. 1**	**Seq. 2**		**Seq. n**
Motif Length	S_0	S_1	S_2	...	S_n

Fig. 2. Individual representation

best results, obtaining the greater hypervolume in seven of the twelve instances tested, and the second best result in four of the remaining five. In these experiments we have also performed an exhaustive statistical study to demonstrate that the differences among the algorithms are statistically relevant. For doing this, we have applied the methodology described in [21], where the authors analyze the sample distributions and the variance homogeneities applying several tests. Depending on the results of these tests, the authors apply parametric or non-parametric tests, always considering a confidence level of 95%. Successful tests are marked with '+' in the last column of Table 2, and negative tests are indicated with '-'. As we can see all the differences are statistically significants. The second comparison uses the CR [7] indicator. Applying this indicator we can compare the nondominated solutions discovered by two algorithms by using the dominance concept. It considers that a covers b if and only if a dominates to b or both belong to the same Pareto front, i. e., $a \succeq b$. Table 3 shows the results of this comparison. If we analyze Table 3 by rows, we see how, again, the second multiobjective adaptation covers a higher percentage of solutions of the other algorithms, achieving an average coverage result of 84.54%. Moreover, if we examine Table 3 by columns, we see how this adaptation (MOABCv2) is the least covered by the other multiobjective adaptations and algorithms with an average coverage percentage of 21.39%. These results demonstrate that MOABCv2 discovers DNA motifs that dominate those discovered by the other algorithms. Moreover, these predictions are not dominated by the predictions made by the other algorithms.

Finally, we have analyzed the motifs discovered by our best multiobjective adaptation (MOABCv2), comparing them with those predicted by fourteen well-known biological methods such as [10]: Consensus, MEME, MEME3, AlignACE, ANN_Spec, Improbizer, MotifSampler, GLAM, SeSiMCMC, Oligo/Dyad-Analysis, MITRA, YMF, QuickScore, and Weeder. Thanks to the methodology defined in [10], we can compare the results obtained by all these biological methods with those obtained by our algorithm. To carry out this comparison, we have used the same biological indicators as in [10]: Sensitivity (nSn), Positive Predictive Value ($nPPV$), Performance Coefficient (nPC), and Corre-

Table 2. Median and IQR of the algorithm hypervolumes

	MOABCv1		MOABCv2		MOABCv3		MOABCv4		MOABCv5		NSGA-II		SPEA2		
Instance	$\bar{HV}$	IQR	$\bar{HV}$	IQR	$\bar{HV}$	IQR	$\bar{HV}$	IQR	$\bar{HV}$	IQR	$\bar{HV}$	IQR	$\bar{HV}$	IQR	
dm01g	79.03%	0.007	83.81%	0.005	83.27%	0.011	83.49%	0.009	83.38%	0.012	81.51%	0.005	83.00%	0.006	+
dm04g	79.17%	0.008	84.31%	0.009	83.62%	0.014	83.71%	0.010	84.04%	0.009	81.09%	0.007	82.28%	0.012	+
dm05g	83.74%	0.006	87.01%	0.007	86.37%	0.006	86.18%	0.007	86.41%	0.010	84.33%	0.006	86.17%	0.007	+
hm03r	51.80%	0.021	61.47%	0.015	60.90%	0.029	58.71%	0.018	61.64%	0.018	47.81%	0.044	52.97%	0.011	+
hm04m	44.71%	0.009	57.66%	0.019	55.60%	0.020	56.34%	0.020	57.04%	0.027	43.57%	0.026	46.48%	0.014	+
hm16g	64.82%	0.024	82.45%	0.036	83.27%	0.061	77.54%	0.037	81.89%	0.045	68.27%	0.018	72.03%	0.012	+
mus02r	54.23%	0.011	65.80%	0.018	64.05%	0.036	65.08%	0.016	64.07%	0.022	59.19%	0.011	59.39%	0.009	+
mus03g	75.24%	0.005	80.05%	0.008	79.65%	0.007	79.85%	0.008	79.74%	0.007	77.19%	0.003	77.56%	0.006	+
mus07g	79.77%	0.011	89.08%	0.042	89.29%	0.019	89.38%	0.010	89.20%	0.037	87.00%	0.005	89.50%	0.005	+
yst03m	62.30%	0.018	70.47%	0.024	68.96%	0.034	71.25%	0.013	69.91%	0.016	65.17%	0.020	66.27%	0.017	+
yst04r	67.37%	0.007	76.28%	0.007	75.19%	0.013	74.47%	0.010	75.58%	0.012	74.77%	0.004	71.46%	0.009	+
yst08r	51.02%	0.011	62.64%	0.024	61.29%	0.029	57.52%	0.017	61.60%	0.018	64.82%	0.011	57.08%	0.021	+

Table 3. Coverage Relation (A $\succeq$ B)

A \ B	MOABCv1	MOABCv2	MOABCv3	MOABCv4	MOABCv5	NSGA-II	SPEA2	average
MOABCv1	-	1.78%	2.02%	2.00%	2.60%	20.92%	12.84%	**10.07%**
MOABCv2	99.23%	-	57.68%	62.80%	86.01%	97.07%	90.83%	**84.54%**
MOABCv3	99.11%	46.83%	-	49.27%	72.87%	93.59%	88.62%	**78.20%**
MOABCv4	99.22%	51.91%	59.27%	-	84.27%	96.62%	89.55%	**82.50%**
MOABCv5	99.11%	23.98%	26.11%	37.67%	-	90.15%	82.97%	**65.11%**
NSGA-II	81.25%	4.56%	6.23%	10.03%	13.39%	-	42.52%	**31.78%**
SPEA2	90.07%	14.16%	17.10%	16.40%	21.68%	60.17%	-	**42.86%**
average	**91.17%**	**21.39%**	**24.98%**	**26.54%**	**41.45%**	**71.34%**	**62.14%**	

lation Coefficient (*nCC*). These biological indicators are calculated with the values of TP (True-Positives), TN (True-Negatives), FP (False-Positives), and FN (False-Negatives), obtained by comparing the set of known binding sites for each instance, with the predictions made by each method (all this information is available in `http://bio.cs.washington.edu/assessment`). Table 4 shows the results of this comparison. To organize the large amount of biological data, and to make this comparison more understandable, first, we have selected the best predictions of the best biological method for each indicator in each instance (information included in the second column of the tables). Then, we have selected the best motif from the set of the nondominated solutions predicted by our algorithm (the motif with the highest value of each indicator). This information is included in the third column of the tables. Finally, in the last column we show

Table 4. Comparison of biological indicators

	Sensitivity (nSn)			**Positive Predictive Value (nPPV)**		
Instance	**best value (method)**	**MOABCv2**	**Increase**	**best value (method)**	**MOABCv2**	**Increase**
dm01g	0.344000 (SeSiMCMC)	**0.472000**	*0.128000*	0.344000 (SeSiMCMC)	**1.000000**	*0.656000*
dm04g	0.022222 (MotifSampler)	**0.392593**	*0.370371*	0.032967 (MotifSampler)	**1.000000**	*0.967033*
dm05g	0.037500 (MEME)	**0.306250**	*0.268750*	0.026667 (MEME)	**1.000000**	*0.973333*
hm03r	0.063726 (MEME)	**0.291667**	*0.227942*	0.108333 (MEME)	**0.660714**	*0.552381*
hm04m	0.005952 (AlignACE)	**0.273810**	*0.267858*	0.006061 (AlignACE)	**0.370370**	*0.364309*
hm16g	0.000000 (-)	**0.390244**	*0.390244*	0.000000 (-)	**0.681818**	*0.681818*
mus02r	0.094828 (MEME)	**0.288793**	*0.193965*	0.142857 (MEME)	**0.761364**	*0.618507*
mus03g	0.281690 (AlignACE)	**0.521127**	*0.239437*	0.256410 (AlignACE)	**1.000000**	*0.743590*
mus07g	0.040000 (ANN_Spec)	**0.540000**	*0.500000*	0.020942 (ANN_Spec)	**1.000000**	*0.979058*
yst03m	0.340136 (Improbizer)	0.272109	*-0.068027*	0.700000 (YMF)	**0.904762**	*0.204762*
yst04r	0.335878 (Consensus)	**0.588785**	*0.252907*	0.357143 (MITRA)	**0.750000**	*0.392857*
yst08r	0.387097 (AlignACE)	0.258065	*-0.129032*	0.786408 (MotifSampler)	0.571429	*-0.214979*

	Performance Coefficient (nPC)			**Correlation Coefficient (nCC)**		
Instance	**best value (method)**	**MOABCv2**	**Increase**	**best value (method)**	**MOABCv2**	**Increase**
dm01g	0.207730 (SeSiMCMC)	**0.414062**	*0.206333*	0.330043 (SeSiMCMC)	**0.629028**	*0.298985*
dm04g	0.013453 (MotifSampler)	**0.279221**	*0.265768*	0.013401 (MotifSampler)	**0.488895**	*0.475494*
dm05g	0.015831 (MEME)	**0.212121**	*0.196290*	0.006491 (MEME)	**0.415052**	*0.408561*
hm03r	0.041801 (MEME)	**0.238710**	*0.196909*	0.063601 (MEME)	**0.414497**	*0.350896*
hm04m	0.003012 (AlignACE)	**0.171875**	*0.168863*	-0.000400 (AlignACE)	**0.291424**	*0.291824*
hm16g	0.000000 (-)	**0.294931**	*0.294931*	-0.005204 (MEME)	**0.458497**	*0.463701*
mus02r	0.060440 (MEME)	**0.264822**	*0.204382*	0.097480 (MEME)	**0.461257**	*0.363777*
mus03g	0.155039 (AlignACE)	**0.418301**	*0.263262*	0.222480 (AlignACE)	**0.605236**	*0.382756*
mus07g	0.013937 (ANN_Spec)	**0.465517**	*0.451580*	0.006056 (ANN_Spec)	**0.649290**	*0.643234*
yst03m	0.261905 (oligodyad)	0.223464	*-0.038441*	0.437304 (oligodyad)	0.373357	*-0.063947*
yst04r	0.202765 (Consensus)	**0.413043**	*0.210278*	0.322430 (Consensus)	**0.581656**	*0.259226*
yst08r	0.269103 (MotifSampler)	0.174174	*-0.094929*	0.470596 (MotifSampler)	0.317704	*-0.152892*

the obtained increases. It is important to note that, while the biological methods only perform well in an specific set of instances (e. g., yeast instances), our proposal discovers good motifs in all of them, regardless of the species. Therefore, we can assume that our multiobjective adaptation will also work well with other kind of instances.

5 Conclusions and Future Work

In this work we demonstrate that the use of a good multiobjective adaptation is important to solve MOPs optimally. To do this, we have defined five multiobjective adaptations of the Artificial Bee Colony (ABC) algorithm to solve an important biological MOP, the Motif Discovery Problem (MDP). After comparing the results obtained by the five multiobjective adaptations among them, and with two standard multiobjective algorithms such as NSGA-II and SPEA2, we have noticed as the second adaptation (Algorithm 2) achieves the best results. In addition, we have compared the motifs discovered by the best ABC multiobjective adaptation with those predicted by fourteen well-known biological methods, checking that our predictions are biologically relevant. As future work, we intend to apply this multiobjective study to other evolutionary algorithms.

Acknowledgments. This work was partially funded by the Spanish Ministry of Science and Innovation and ERDF (the European Regional Development Fund), under the contract TIN2008-06491-C04-04 (the M* project). Thanks also to the Fundación Valhondo, for the economic support offered to David L. González-Álvarez.

References

1. Deb, K.: Multi-objective optimization using evolutionary algorithms. John Wiley & Sons (2001)
2. Fogel, L.J.: Artificial Intelligence Through Simulated Evolution. Forty Years of Evolutionary Programming. John Wiley & Sonc, Inc., New York (1999)
3. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Technical report-tr06, Erciyes University, Turkey (2005)
4. D'haeseleer, P.: What are DNA sequence motifs? Nature Biotechnology 24(4), 423–425 (2006)
5. Fonseca, C.M., Fleming, P.J.: Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In: Proceedings of the 5th International Conference on Genetic Algorithms, San Francisco, CA, USA, pp. 416–423 (1993)
6. While, L., Hingston, P., Barone, L., Huband, S.: A faster algorithm for calculating hypervolume. IEEE Transactions on Evolutionary Computation 10(1), 29–38 (2006)
7. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: empirical results. Evolutionary Computation 8(2), 173–195 (2000)

8. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6, 182–197 (2002)
9. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength pareto evolutionary algorithm. Technical report tik-report 103, Swiss Federal Institute of Technology Zurich, Switzeland (2001)
10. Tompa, M., et al.: Assessing computational tools for the discovery of transcription factor binding sites. Nature Biotechnology 23(1), 137–144 (2005)
11. Liu, F.F.M., Tsai, J.J.P., Chen, R.M., Chen, S.N., Shih, S.H.: FMGA: Finding motifs by genetic algorithm. In: Fourth IEEE Symposium on Bioinformatics and Bioengineering (BIBE 2004), pp. 459–466 (2004)
12. Stine, M., Dasgupta, D., Mukatira, S.: Motif discovery in upstream sequences of coordinately expressed genes. In: The 2003 Congress on Evolutionary Computation (CEC 2003), vol. 3, pp. 1596–1603 (2003)
13. Che, D., Song, Y., Rashedd, K.: MDGA: Motif discovery using a genetic algorithm. In: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation (GECCO 2005), pp. 447–452 (2005)
14. Shao, L., Chen, Y.: Bacterial foraging optimization algorithm integrating tabu search for motif discovery. In: IEEE International Conference on Bioinformatics and Biomedicine (BIBM 2009), pp. 415–418 (2009)
15. Shao, L., Chen, Y., Abraham, A.: Motif discovery using evolutionary algorithms. In: International Conference of Soft Computing and Pattern Recognition (SOCPAR 2009), pp. 420–425 (2009)
16. Fogel, G.B., et al.: Discovery of sequence motifs related to coexpression of genes using evolutionary computation. Nucleic Acids Research 32(13), 3826–3835 (2004)
17. Fogel, G.B., et al.: Evolutionary computation for discovery of composite transcription factor binding sites. Nucleic Acids Research 36(21), e142, 1–14 (2008)
18. Kaya, M.: MOGAMOD: Multi-objective genetic algorithm for motif discovery. Expert Systems with Applications 36(2), 1039–1047 (2009)
19. Wingender, E., Dietze, P., Karas, H., Knuppel, R.: TRANSFAC: a database on transcription factors and their DNA binding sites. Nucleic Acids Research 24(1), 238–241 (1996)
20. González-Álvarez, D.L., Vega-Rodríguez, M.A., Gómez-Pulido, J.A., Sánchez-Pérez, J.M.: Finding Motifs in DNA Sequences Applying a Multiobjective Artificial Bee Colony (MOABC) Algorithm. In: Giacobini, M. (ed.) EvoBIO 2011. LNCS, vol. 6623, pp. 89–100. Springer, Heidelberg (2011)
21. Sheskin, D.J.: Handbook of parametric and nonparametric statistical procedures, 4th edn. Chapman & Hall/CRC Press, New York (2007)

The Role of Mutations in Whole Genome Duplication

Qinxin Pan, Christian Darabos, and Jason H. Moore

Computational Genetics Laboratory,
Dartmouth College, Hanover, NH 03755, USA
Qinxin.Pan@dartmouth.edu
http://www.epistasis.org/

Abstract. Genetic mutation is an essential factor in the evolution of biological organisms and a driving force of phenotypical innovation. On rare occasions, nature takes a major evolutionary leap during which an organism's gene repertoire suddenly doubled. Genetic mutation affects both the whole genome duplication as it happens, and also during all the subsequent evolutionary steps. We develop a Boolean model of gene regulatory networks that simulates the duplication event and subsequent Darwinian evolution using an evolutionary algorithm. We analyze the role of these two different types of mutations on synthetic systems. Our results show that high *duplication mutation* rate triggers the development of new phenotypes, advantageous in a changing environment, to the detriment of environmental robustness. Additionally, our research highlights the necessity of a low *evolutionary mutation* rate for the survival of duplicated individuals within a mixed population, ensuring the spreading novel phenotype. We conclude that both types of mutations play complementary roles in determining the successful propagation of organisms with duplicated genomes.

Keywords: Whole Genome Duplication, duplication mutations, evolutionary mutations, fitness, robustness.

1 Introduction

Evolution is nature's way of giving living organisms the opportunity to develop new abilities and adapt to their (changing) environment. It is at the origin of the diversity at every level of biological organization, from species, to individual organisms, to the genes themselves. Evolution happens across successive generations through three main mechanisms: natural selection, recombination and mutation. Throughout millennia, evolution has been constantly operating at a slow pace, steadily increasing the robustness and fitness of biological organisms. In rare cases, evolution takes a giant leap, duplicating parts or the totality of a genome, usually in response to drastic environmental changes.

Studies have examined the effect of evolutionary mutation rate on the speed of adaptive evolution and introduce the idea of "survival of flattest", where high

M. Giacobini, L. Vanneschi, and W.S. Bush (Eds.): EvoBIO 2012, LNCS 7246, pp. 122–133, 2012.

evolutionary mutation rate tends to select the genotype with flatter regions of the fitness landscape although they occupy lower fitness peaks [21]. However, the effect of mutation during major evolutionary jumps, and during the following evolutionary steps, although well established phenomena, remain open questions. In a preliminary work [14], we proposed a Boolean model that simulates whole genome duplication (WGD) in simple biological organisms, and studied the effect of subsequent diversification. In the present article, we build on that model, refining it to mimmic even more closely the behavior observed in biology, and address the role of the two different types of mutations: the duplication mutation (DM) that happens once during the duplication process, and evolutionary mutation (EM) happening at every generation. Using Random Boolean Networks (RBNs) [10] to model the dynamics of gene regulatory networks (GRNs), we simulate WGD events in simple synthetic biological organisms. We separately quantify the effect of DM and EM on environmental robustness and evolutionary innovation. Then, we investigate the effect of EM on the survival of duplicated and non-duplicated individuals in a mixed population. We conclude by discussing the results and offering possible future research directions.

2 Background

At the gene level, a regulatory network, or GRN, is made of a set of genes, as vertices, linked by gene-products (protein, mRNA, miRNA), as directed edges, representing the regulatory influence of a source gene on a target gene. Though extremely complex, by abstracting number of the particular kinetics of the biochemical interactions, one can still study the global dynamics of GRNs. Random Boolean networks (RBNs), where N vertices represent the binary (*on/off*) expression state of the genes, have been extensively used to model the temporal changes in simple GRNs [10]. The expression of each gene is governed by a randomly generated Boolean function of its upstream gene(s) with probability p_{expr}. Every vertex changes its expression state instantaneously, synchronously, and in discreet time steps. The set of all expression states of all genes at any given time t is called a *configuration*. As the system is finite, there are 2^N possible configurations. Therefore, starting in an arbitrary initial configuration, the system deterministically travels through a sequence of transient configurations. It will eventually encounter a previously visited configuration, thus entering a limit cycle called an *attarctor*.

By tuning the input and output degree distributions of the underlying directed network and the expression probability within the Boolean function, the RBN dynamics undergoes a phase transition from an *ordered* regime, with short and stable attractors, to *chaotic* regime, with longer attractors that are more sensitive to small perturbations. According to Kauffman's conjecture, biological organisms operate in the ordered regime, at the edge of chaos, in a region called *critical*. The critical regime offers a tradeoff between the ability to withstand environmental perturbations (robustness) and the ability to utilize these perturbations for evolutionary innovation (evolvability) [2]. Regardless of the level of

biological organization, living organisms display remarkable resilience to changing conditions, and at the same time, they are able to respond to these changes by developing novel phenotypes. At first glance, these qualities seem paradoxical, yet both empirical [7,8] and theoretical [2,20] analyses suggest that they are, in fact, complementary.

When a more rapid, or more profound response is made necessary by sudden and extreme environmental perturbations, biological organisms may undergo a drastic evolutionary process in the form of a whole genome duplication (WGD). During a WGD, the entire gene repertoire of an organism, including the regulatory interactions, is doubled [16]. WGD has long been recognized as a driver of evolutionary innovation [12] and recent genetic analyses have demonstrated several major evolutionary transitions resulted from ancient WGD events [11,6,17]. The duplication of genetic materials has implications for environmental robustness, as redundant genes diverge to compartmentalize the original function of the ancestral gene (subfunctionalization) [16]. In *S. cerevisiae*, for example, this occurs through the differential expression of redundant genes under various growth conditions [9]. WGD also has implications for evolutionary innovation, as duplicate genes diverge to acquire new functions (neofunctionalization) [16]. In *S. cerevisiae*, the ability to consume glucose and grow anaerobically have both been attributed to the genetic diversification that followed a WGD event [15].

Immediately after undergoing a WGD event, the stability, and thus the fitness of the new phenotypes is generally greatly reduced [19]. Because a non-duplicated organism is supposedly optimally adapted to its (ancestral) environment, WGD is highly detrimental to the organism, and is oftentimes fatal. When such a large quantity of genetic material is produced at once, the number of transcriptional errors increases dramatically, leading to phenotypes ill adapted to their ancestral environment. We call this the duplication mutation (DM). However, duplicate genes also supply new genetic material, which can be shaped via evolutionary mutation (EM) and selection to produce novel functions. These functions may allow for more rapid adaptation if a new environment is encountered, providing potential fitness benefits [19].

3 Methods

3.1 RBN Topology

In RBN, the topology of the underlying network has an crucial influence on system dynamics [3,13,2]. Empirical evidence suggest that, unlike Kauffman's original random connectivity, the output degree distributions of GRNs of several organisms is highly inhomogeneous [3,1]. There is a small number of hubs with high outgoing connectivity, and the vast majority of the nodes are, on the contrary, sparsely connect. Thus, the outgoing degree distribution is *heavy tailed*, following a power-law distribution $p \sim ax^{-\gamma}$. The incoming degree distribution is much more homogeneous, following a Poisson distribution centered around the network's average degree $\bar{k}$. Our RBN topologies are generated accordingly, as described by [5]. We consider RBNs with $N = 10$ nodes prior to duplication and

$N = 20$ nodes after duplication due to the high computational cost. RBNs are initialized near the most biological relevant regime by setting the probability of gene expression within the look-up table p_{expr} to 0.5 and the scaling exponent γ to 1.894 [2].

3.2 Genome Wide Duplication and Duplication Mutation

In our synthetic systems, unlike in nature, we are able to simulate a *perfect* WGD event that does not change its dynamical behavior. First, we create a mirror-image of the original RBN. Then, duplicate and original components are linked by new edges from the source nodes in one component to the targets in the other (Fig. 1). After duplication, each node has twice as many inputs and outputs as the corresponding node in the original (non-duplicated) system. As a result, the number of entries in the look-up table is squared. Supplementary entries are populated to simulate redundancy. This mimics the biological scenario, when the original gene or it duplicate is expressed, the gene's product is present in the medium; otherwise, if both are repressed, original gene's product is absent.

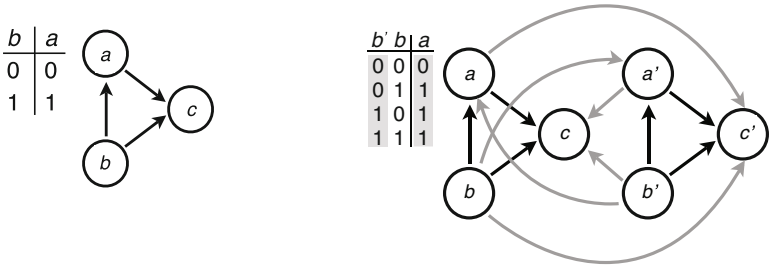


Fig. 1. Duplication of RBNs: (a) original (non-duplicated) system and (b) duplicated system. After duplication, each node has twice as many edges. The black edges indication the pre-existing edges whereas the grey edges represent the new edges. To expand the Boolean rule table, we mimic the original biochemical influences. For example, when b and b' are both *repressed*, the function is identical to b is *repressed* in the original rule; when at least one of b and b' is *expressed*, we consider that the b's gene product is present in the medium.

Finally, to simulate the imperfect WGD in a biological organisms, we introduced DM in the Boolean function. With a probability M_D (also called DM rate), entries in the Boolean rules are flipped. We explore in detail the effect of a wide range of M_D values in Section 4.

3.3 Environmental Robustness

Environmental perturbations come in many forms, including alterations in temperature, growth medium, or biotic environment. A RBN is environmentally

robust if its phenotype is insensitive to these non-genetic perturbations. We measure environmental robustness as the sensitivity of a RBN to the perturbation of a single, randomly chosen configuration of its attractor. Specifically, we systematically perturb the state of each node in the randomly chosen configuration, one at a time, and measure the proportion of perturbations in which the RBN returns to its original attractor.

3.4 Evolutionary Innovation

An evolutionary innovation can be thought of as a change in phenotype that confers a fitness advantage. In our system, the phenotype is represented by the attractor. To assess evolutionary innovation, we measure the fitness of a RBN as the ability of its attractor to match a target attractor. This target attractor represents the gene expression pattern required for optimal adaptation to a given environment [13]. Fitness thus provides a proxy for evolutionary innovation.

For each RBN, we randomly select a single output node and record the sequence of output states σ_{out} while the system is cycling in the attractor. The fitness F of a RBN is then calculated as the Hamming distance between the output and target sequences,

$$F = \max \left\{ 1 - \frac{1}{\text{lcm}(L, L_c)} \sum_{t=1}^{\text{lcm}(L,L_c)} |\sigma_{\text{out}}(t) - \sigma_{\text{target}}(t)| \right\}, \tag{1}$$

where L is the length of the output sequence, L_c is the length of the target sequence, and lcm denotes the least common multiple. To facilitate the comparison of sequences with $L \neq L_c$, both sequences are concatenated onto themselves until they are of length $\text{lcm}(L, L_c)$. To ensure that fitness is independent of the starting position of the output sequences, we take the maximum fitness over all cyclic permutations of σ_{out}. All fitness reported in this article are average of the whole population.

3.5 Evolution and Evolutionary Mutation

We simulate the evolution of randomly initialized populations of 100 RBNs, each is paired with its own, randomly chosen Boolean function in the form of a lookup table, and initial state which do not change throughout the evolutionary trajectory of its lineage. Each population is evolved for 1000 to 2000 discrete, non-overlapping generations, and experiments are replicated independently 100 times. In every generation, the fitness of each RBN is assessed according to Eq. 1. RBNs are then selected with uniform probability, with replacement, to compete in binary tournaments. Within a tournament, the RBN with the highest fitness is selected to move on to the next generation, after undergoing evolutionary mutation. EM only affects the RBN's look-up tables, such that the entries in the look-up tables associated with each vertex undergo bit-flip mutation with probability M_E. A wide range of different M_E values will be explored in the experimental part in Section 4. This process of selection and mutation is repeated until the next generation is fully populated.

4 Experimental Results

4.1 Duplication Mutations in an Ancestral Environment

To simulate an ancestral environment, we assume the original RBNs are optimally adapted to their environment. We use this optimal phenotype (single gene expression sequence) as the target to evaluate the fitness of the duplicated systems. When measure the fitness immediately after a perfect duplication, $M_D = 0$, the RBNs display the exact same attractors as their non-duplicated counterparts. Therefore, they maintain an optimal fitness $F = 1.0$ in the ancestral environment (left-most point in Fig. 2).

However, once we introduce DM, the fitness decreases rapidly as M_D increases. Figure 2 shows the results of these experiments in a log-lin scale.

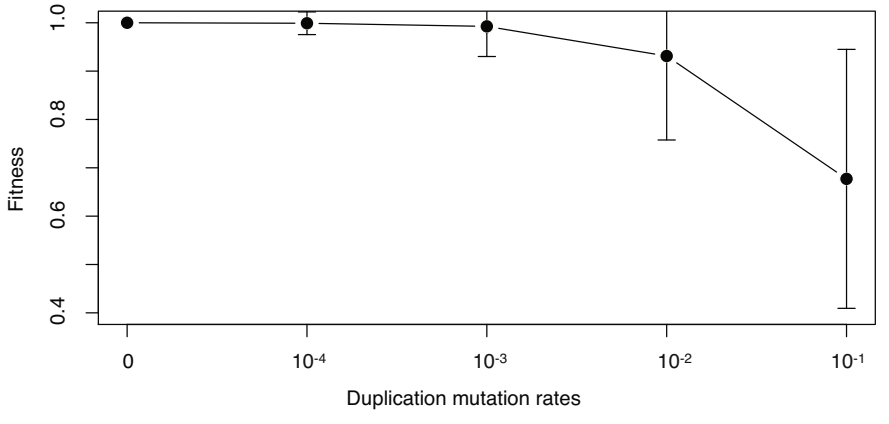


Fig. 2. Effect of DM on instant fitness in ancestral environment. The average fitness and standard deviation are reported for 10,000 independent repetitions.

Even though perfectly the duplicated RBNs exhibit the same phenotype as the non-duplicated, the duplication event proceeds to shift the gene expression p_{expr} away from 0.5 (left and center panels of Fig. 3c). This phenomenon makes the Boolean functions more canalizing, which in turn tilts our systems closer towards the ordered regime with attractors that are short and stable. The redundancy in the perfectly duplicated RBNs and the canalization of the rules lead to an increase of environmental robustness (two left-most points in Fig. 3a).

Once we introduce DMs in the system, p_{expr} move towards 0.5 (right panel of Fig. 3c). Now, the RBNs shifts closer to the chaotic regime as the DM rate increases, with longer, less stable attractors. Consequently, we observe a decrease in the environmental robustness in Fig. 3a and an increase in attractor length in Fig. 3b as the DM rate increases.

These results are consistent with the negative effect of WGD observed in biological organisms where the duplication is imperfect and detrimental to the organism in its ancestral environment.

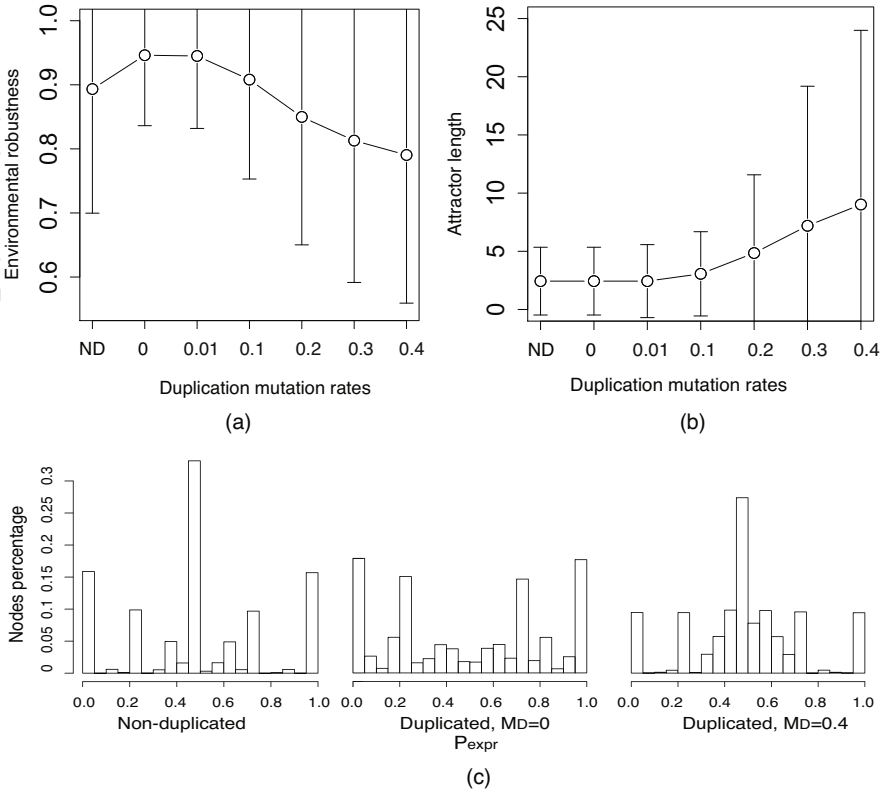


Fig. 3. Effect of DM on: (a) robustness, (b) attractor length and (c) distribution of P_{expr}. Results depicts 10000 replicates.

4.2 Effects of Mutations on Evolutionary Innovation

To understand the effect of mutations on the evolutionary innovation capabilities of RBNs, we have to present both original and duplicated systems with a novel environment they must adapt to. Consequently, we produce a random phenotype in the form of a gene expression pattern of length 10 that becomes the target sequence the output gene of our systems must evolve towards. We investigate separately the effect of DM (thus fixing the EM to a constant rate) and the effect of EM (fixing the rate of DM).

Duplication Mutation: As we focus exclusively on DM, we fix the EM rate to $M_E = 0.002$ for the rest of this section. Perfectly duplicated RBNs achieve higher fitness than original systems during the evolution process (Fig. 4a). However, this advantage comes at the cost of environmental robustness (Fig. 4b). Imperfect duplicates achieve even higher fitness. RBNs show marginally longer attractors and lower robustness with the increasing of M_D (Fig. 3b and inset). We explore

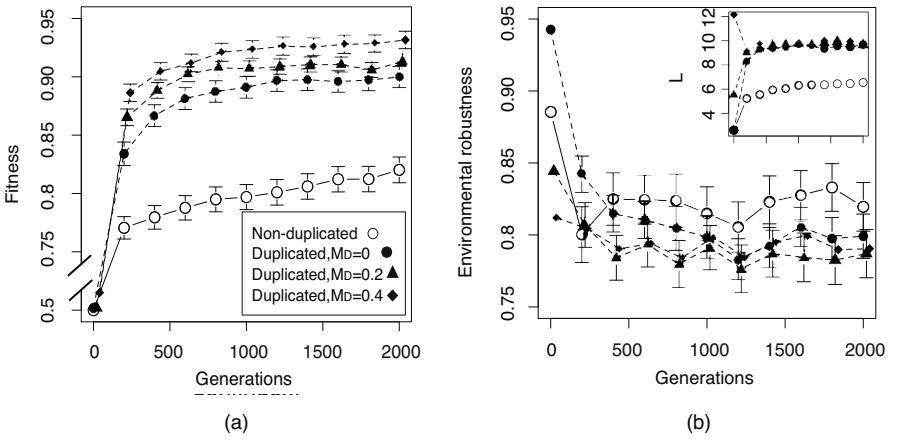


Fig. 4. Effect of DM in evolution on: (a) fitness and (b) environmental robustness of RBNs with $M_D \in [0, 0.2, 0.4]$ under fixed evolutionary mutation rate (M_E=0.002). Inset in panel (b) depicts attractor length change during evolution. Mean and standard deviation are reported for 100 repeats. Scale of x axis is identical in all panels, including the inset.

$M_D \in [0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 0.2, 0.3, 0.4]$. For readability reasons, we only show results for more extreme values. The trends are however similar.

These results are consistent with the finding that WGD is beneficial in a novel environment. Moreover, besides the duplication itself, the DM further increase the fitness a population can reach in evolution at the cost environmental robustness.

Evolutionary Mutations: From a biological viewpoint, successful WGD events that produce viable phenotypes are extremely rare. In this section, we focus on the effect on EM, and fix the duplication mutation rate to M_D=0.1 which is believed to be orders of magnitude higher than the evolutionary mutation rate [16]. We explore a wide range of evolution mutation rates $M_E \in [10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}]$ and report the results after 1000 generations in Fig. 5. We observe that although the extreme values of M_E do not cause significant difference in fitness between different individuals, there is a transition in the intermediate values, where duplicated systems yield higher fitness than non-duplicated ones for $M_E = 10^{-3}$ whereas this trend is inverted for $M_E = 10^{-2}$ (Fig. 5a). Regardless of the EM rate, non-duplicated RBNs maintain a more constant, and generally higher environmental robustness (Fig. 5b).

To understand why different EM rates influence different RBNs differently, we need to examine how robust the systems are to genotypical mutations. Thus we measure the *mutational* robustness, which is the insensitivity of the RBNs phenotype to the genetic perturbations. Mutational robustness is evaluated by flipping each entry of its Boolean rules with a probability p_m and measuring whether the system fall into the same attractor before and after the perturbation [20]. Values reported in Fig. 6b are averaged over 1000 different perturbations

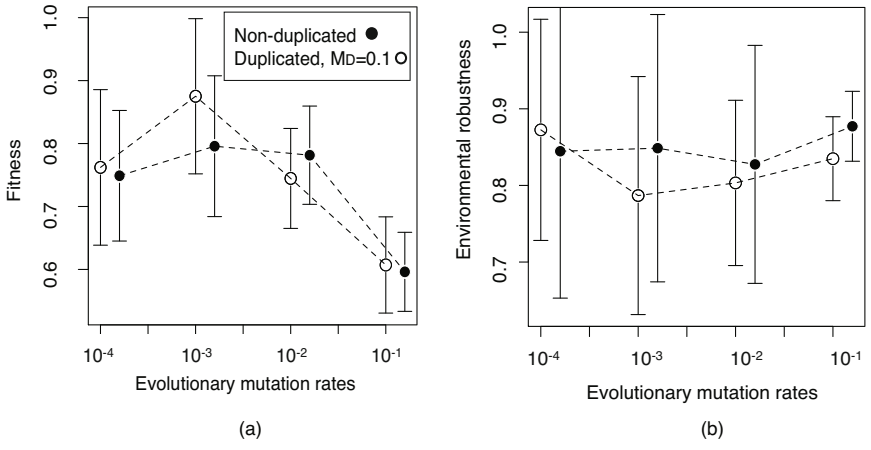


Fig. 5. Effect of EM in evolution: (a) fitness of RBNs at generation 1000 and (b) environmental robustness at generation 1000. Values are reported as the average and standard deviation over 100 runs.

on RBNs at the end of the evolution ($M_E = 10^{-3}$). In addition, we measure the fitness of these genetically perturbed RBNs and report those values in Fig. 6a.

When the Boolean rules are perturbed, the original RBNs always show higher mutational robustness than the duplicated ones (Fig. 6b). This leads to a flatter fitness landscape of the original RBNs than that of the duplicated ones in Fig. 6a.

Therefore, although the duplicated RBNs generate fitter phenotypes at $M_E = 10^{-3}$, they are also more fragile to genotypical mutations. When the EM rate is higher, systems are unable to maintain the fitness during evolution. Oppositely, the non-duplicated models are more robust to genotypical mutations and thus maintain the fitness better under high EM rate.

4.3 Survival of the Duplicated RBNs

The difference in fitness between the original and the duplicated RBNs over the spectrum of EM rates raises the question whether the dominance of duplicated systems is dependent on the EM rate. Specifically, if WGD only affects a subset of the population at generation zero, whether the EM rate changes which of the duplicate or original individuals will dominate the population at the end of the evolutionary process. To answer this question, we constructed mixed populations with an equal number of non-duplicated and duplicated RBNs. We evolve them under different EM rates. We observe a phase transition reported in Fig.7. When the EM rate M_E=10^{-3}, the duplicated RBNs take over the entire population in $\sim$ 70% of the repetitions. Whereas when M_E=10^{-2}, it is the non-duplicated

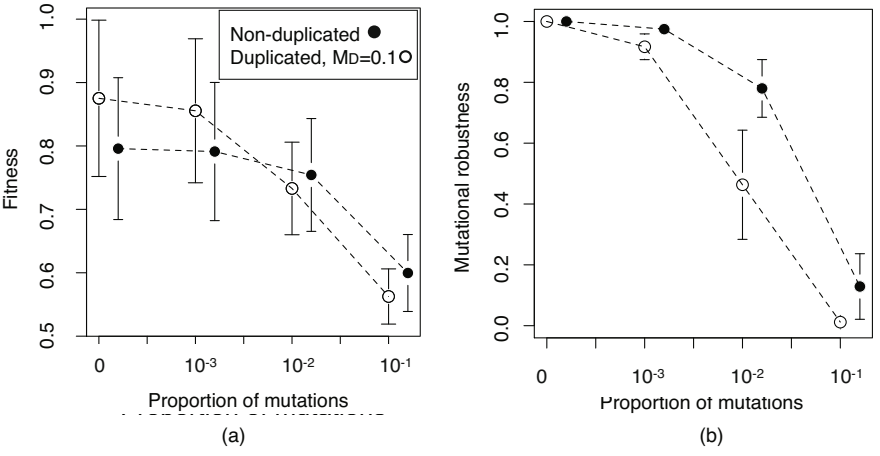


Fig. 6. Comparision of different RBNs after evolution: (a) Fitness of RBNs genetically perturbed at generation 1000 over $p_m \in [0, 10^{-3}, 10^{-2}, 10^{-1}]$, (b) mutational robustness after evolution on the same range of p_m. Results reported are averages and standard deviations over 1,000 samplings.

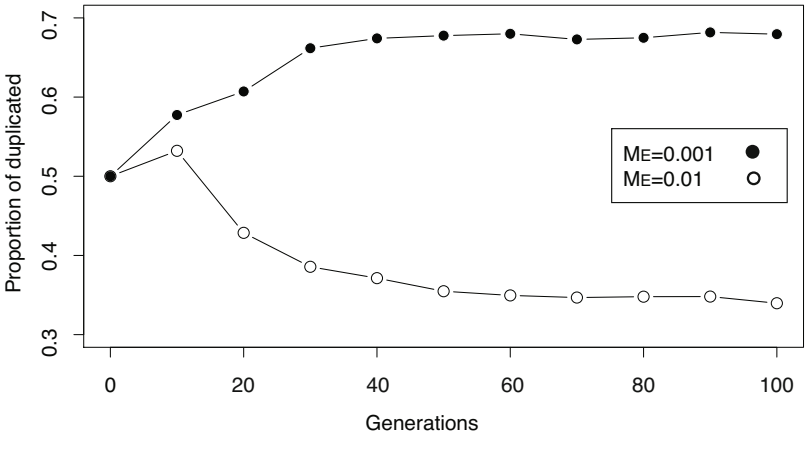


Fig. 7. Effect of EM rate on the survival of duplicated RBNs: proportion of duplicated RBNs within the mixed populations during evolution is reported. Results are averaged over 1000 repetitions.

RBNs that overtake in the same proportion. This is a prime example of the "survival of flattest", where individual with generally lower, but more constant fitnesses get selected over those with higher but narrow fitness peaks [21].

5 Discussion

We have used Random Boolean Networks (RBNs) to investigate the role of two different, yet complementary types of mutations in whole genome duplication (WGD) event. Despite their abstract nature, RBNs are able to closely model and predict a wide range of cellular and molecular phenomena [18]. Because our model displays the same attractor before and after a perfect duplication, we believe it is apt to simulate the WGD event. Our analysis helps clarify the roles of duplication mutations and evolutionary mutations in WGD. Specifically, we show that although perfect duplication does not change the phenotype and actually increases the environmental robustness, the duplication mutations during WGD are detrimental in ancestral environment. On the contrary, duplication mutations provide marginal benefit in novel environments, at the expense of environmental robustness. Over evolutionary time, these differences magnified, with duplicated RBNs achieving significantly higher fitness and lower environmental robustness than the non-duplicated counterparts. The duplication mutations further increases the fitness and environmental robustness gap between original and duplicated RBNs.

The trade-off between fitness and environmental robustness of both systems is clear in mixed populations, and highlighted by the phase transition at different evolutionary mutation rates. Consistent with the "survival of flattest", the non-duplicated RBNs with a higher mutational robustness thrive at higher evolutionary mutation rate whereas the duplicated RBNs with a lower mutational robustness are taking over the population at lower evolutionary mutation rates.

In conclusion, using simple abstract models of GRN, this work sheds some light on WGD events in biological organisms. It highlights the necessity for a high duplication mutation rate that triggers the development of new phenotypes, though lowering their robustness. It also explains a low evolutionary mutation rate, crucial not only because it improves fitness over time, but also because it ensures the domination of duplicated individuals in the whole population.

Future work will seek to test our model with different parameters and incorporate diversification. Studies shown that rapid diversification via gene loss and edge rewiring happen immediately after WGD [16]. The advantages that the diversification can bring to the system in terms of fitness and robustness will be helpful to understand this process. Moreover, several studies shown that transcription factors, which are usually hubs in GRN, are better maintained than other genes after WGD [4]. It will be insightful to look at that bias of gene loss from the evolvability and robustness view.

Acknowledgments. This work was partially supported by NIH grants LM009012, LM010098, AI59694, and by the Swiss National Science Foundation grant PBLAP3-136923.

References

1. Albert, R.: Scale-free networks in cell biology. Journal of Cell Science 118, 4947–4957 (2005)
2. Aldana, M., Balleza, E., Kauffman, S., Resendiz, O.: Robustness and evolvability in genetic regulatory networks. Journal of Theoretical Biology 245, 433–448 (2007)
3. Aldana, M., Cluzel, P.: A natural class of robust networks. Proceedings of the National Academy of Sciences 100, 8710–8714 (2003)
4. Brunet, F.G., Crollius, H.R., Paris, M., Aury, J.M., Gibert, P., Jaillon, O., Laudet, V., Robinson-Rechavi, M.: Gene loss and evolutionary rates following whole-genome duplication in teleost fishes. Molecular Biology and Evolution 23(9), 1808–1816 (2006)
5. Darabos, C., Tomassini, M., Giacobini, M.: Dynamics of unperturbed and noisy generalized Boolean networks. Journal of Theoretical Biology 260, 531–544 (2009)
6. De Bodt, S., Maere, S., Van de Peer, Y.: Genome duplication and the origin of the angiosperms. TRENDS in Ecology and Evolution 20, 591–597 (2005)
7. Ferrada, E., Wagner, A.: Protein robustness promotes evolutionary innovations on large evolutionary time-scales. Proceedings of the Royal Society London B 275, 1595–1602 (2008)
8. Isalan, M., Lemerle, C., Michalodimitrakis, K., Horn, C., Beltrao, P., Raineri, E., Garriga-Canut, M., Serrano, L.: Evolvability and hierarchy in rewired bacterial gene networks. Nature 452, 840–846 (2008)
9. Kafri, R., Bar-Even, A., Pilpel, Y.: Transcription control reprogramming in genetic backup circuits. Nature Genetics 37, 295–299 (2005)
10. Kauffman, S.A.: Metabolic stability and epigenesis in randomly constructed genetic nets. Journal of Theoretical Biology 22, 437–467 (1969)
11. Kellis, M., Birren, B.W., Lander, E.S.: Proof and evolutionary analysis of ancient genome duplication in the yeast Saccharomyces cerevisiae. Nature 428, 617–624 (2004)
12. Ohno, S.: Evolution by Gene Duplication. George Allen and Unwin, London (1970)
13. Oikonomou, P., Cluzel, P.: Effects of topology on network evolution. Nature Physics 2, 532–536 (2006)
14. Pan, Q., Darabos, C., Tyler, A.L., Moore, J.H., Payne, J.L.: The influence of whole genome duplication and subsequent diversification on environmental robustness and evolutionary innovation in gene regulatory networks. In: Proceedings of the European Conference on Artificial Life, pp. 614–621 (2011)
15. Piškur, J.: Origin of the duplicated regions in the yeast genomes. Trends in Genetics 17, 302–303 (2001)
16. Sémon, M., Wolfe, K.H.: Consequences of genome duplication. Current Opinion in Genetics and Development 17, 505–512 (2007)
17. Taylor, J.S., Braasch, I., Frickey, T., Meyer, A., Van de Peer, Y.: Genome duplication, a trait shared by 22,000 species of ray-finned fish. Genome Research 13, 382–390 (2003)
18. Thakar, J., Pilione, M., Kirimanjeswara, G., Harvill, E.T., Albert, R.: Modeling systems-level regulation of host immune responses. PLoS Comput. Biol. 3(6), e109 (2007)
19. van Hoek, M.J.A., Hogeweg, P.: Metabolic adaptation after whole genome duplication. Molecular Biology and Evolution 26, 2441–2453 (2009)
20. Wagner, A.: Robustness and evolvability: a paradox resolved. Proc. Biol. Sci. 275(1630), 91–100 (2008)
21. Wilke, C.O., Wang, J.L., Ofria, C., Lenski, R.E., Adami, C.: Evolution of digital organisms at high mutation rates leads to survival of the flattest. Nature 412, 331–333 (2001)

Comparison of Methods for Meta-dimensional Data Analysis Using in Silico and Biological Data Sets

Emily R. Holzinger[1], Scott M. Dudek[1], Alex T. Frase[2], Brooke Fridley[3], Prabhakar Chalise[3], and Marylyn D. Ritchie[2]

[1] Center for Human Genetics Research, Vanderbilt University, Nashville, TN, USA
emily.r.holzinger@vanderbilt.edu, dudek@chgr.mc.vanderbilt.edu
[2] Center for Systems Genomics, Pennsylvania State University, University Park, PA, USA
{alex.frase,marylyn.ritchie}@psu.edu
[3] Divison of Biomedical Statistics and Informatics, Mayo Clinic College of Medicine, Rochester, MN, USA
fridley.brooke@mayo.com, pchalise@kumc.edu

Abstract. Recent technological innovations have catalyzed the generation of a massive amount of data at various levels of biological regulation, including DNA, RNA and protein. Due to the complex nature of biology, the underlying model may only be discovered by integrating different types of high-throughput data to perform a "meta-dimensional" analysis. For this study, we used simulated gene expression and genotype data to compare three methods that show potential for integrating different types of data in order to generate models that predict a given phenotype: the Analysis Tool for Heritable and Environmental Network Associations (ATHENA), Random Jungle (RJ), and Lasso. Based on our results, we applied RJ and ATHENA sequentially to a biological data set that consisted of genome-wide genotypes and gene expression levels from lymphoblastoid cell lines (LCLs) to predict cytotoxicity. The best model consisted of two SNPs and two gene expression variables with an r-squared value of 0.32.

Keywords: Systems biology, neural networks, evolutionary computation, data integration, human genetics.

1 Introduction

1.1 A Systems Biology Approach for Complex Genetic Traits

A major focus of recent human genetics research has been to uncover the etiology of common, complex phenotypes. Over the past decade genome-wide association studies (GWAS) have discovered thousands of SNPs that significantly associate with hundreds of complex traits [1]. However, the effect sizes of the significant SNPs are usually tiny and collectively they only explain a small portion of the estimated variability in phenotype due to genetic factors [2]. These results are not surprising if one considers that an extremely simplistic study design is being used to study complex biological processes.

M. Giacobini, L. Vanneschi, and W.S. Bush (Eds.): EvoBIO 2012, LNCS 7246, pp. 134–143, 2012.

One key to understanding the complexity that underlies traits will be effectively integrating different types of data, such as genome-wide gene expression levels and SNP genotypes [3]. Tools for an effective data integration analysis must be able to perform several key tasks. First, they need to be able to handle both quantitative and categorical predictor variables. Next, they should be able to deal with the inherently high level of noise in these data sets to perform accurate variable selection [4]. Finally, these methods should be able to integrate the different data types to form "meta-dimensional" predictive models. The prefix meta- refers to the fact that these models will encompass different kinds of multi-dimensional models. While no single method to date is able to perform all of these tasks seamlessly, there are several candidates that show potential for testing systems biology-based hypotheses to elucidate the etiology of complex phenotypes.

For this study, we chose to test three analysis methods using simulated, or in silico, meta-dimensional data – Lasso, Random Jungle, and evolutionary computation methods within ATHENA. These methods were selected as they are capable of performing meta-dimensional analyses on high-throughput data while using different types of algorithms.

2 Methods

2.1 Data Simulation

To test these methods, we modified a previously developed simulation technique [5] to generate SNP genotype and gene expression variables (EVs) that predict a quantitative outcome. Each model was simulated with 3 functional variables: 2 SNPs and 1 EV. In total 10 models were generated: four with only main effects of the three variables, four with an interaction effect between the two SNPs, and two null data sets with no functional variables.

SNP genotype data was randomly generated using genomeSIMLA [6]. The data was simulated with patterns of correlation for 100 or 1000 SNPs to represent the naturally occurring linkage disequilibrium. The two functional SNPs were selected to have a minor allele frequency (MAF) of 0.3. A total of 1000 data sets were generated (100 for each of the 10 models). Genotype data was generated for 500 individuals / data set.

Gene expression data was simulated using a multivariate random normal distribution (MVN) with forced correlation between specific SNPs and EVs. For a given EV (X) and individual (i), the distribution was defined as $X \sim MVN(\mu_i, \sum X)$, where the mean μ_i is calculated from the product of effect matrix B and the vector of SNP genotypes G for individual i so that: $\mu_i = G_i * B$. The effect matrix is shown in Figure 1. The number of rows and columns are equal to the number of SNPs and EVs, respectively, and k is the correlation between a given SNP and EV. This correlation is modeling the occurrence of expression quantitative loci (eQTLs) where SNPs are associated with an expression trait. For this analysis, we

$$B = \begin{bmatrix} k & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & k & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & k \end{bmatrix}$$

Fig. 1. Effect matrix B showing the relationship between SNPs (rows) and EVs (columns)

set k = 0.8, which results in an r-squared value of ~0.3-0.35 between the SNP and EV for all data sets. Biologically speaking, k is likely to be variable across SNP-EV pairs. This layer of complexity should be added into the simulation for studies that are exploring eQTL detection specifically.

The covariance matrix $\sum X$ used to generate the MVN was calculated from the observed correlation structure of 50 or 500 real EVs selected at random from a data set downloaded from the Gene Expression Omnibus (GEO) website [7]. The data set consisted of transformed and normalized microarray data for baseline EV levels of LCLs in the study described in [8] (Accession Number: GSE7792).

The quantitative outcome was generated to represent log transformed IC50 values from a cytotoxicity study using LCLs where IC50 is the concentration of drug at which 50% of the cells remain viable. The outcome variables were selected from a normal distribution generated for each individual with standard deviation (sd) of 0.56. The sd was calculated from the real log IC50 values generated by the same study used in step 2. These values were downloaded from PharmGKB website [9] (Accession Number: PS206922). The mean of the distribution (μ_i) was calculated the values of the functional variables for each individual and coefficients determined to produce a given effect size (Eq. 1). An example of a main effect (Eq. 2) calculation is shown below:

$$\mu_i = a + b_1(X_1) + b_2(X_2) + b_3(X_3) \quad (1)$$

$$\mu_i = -0.5 + 0.3(SNP.1) + 0.3(SNP.2) + 0.4(EV.1) \quad (2)$$

Table 1. Description of simulated data sets (100 data set /model). Mean (standard deviation) for adjusted r-squared values calculated from: [1]Univariable linear regression analyses, [2]Multivariable linear regression analyses that included all direct main and inte terms, and [3]Multivariable linear regression analyses that included only direct main effects.

Variable count (SNP/EV)	Effect Type	SNP 1[1]	SNP 2[1]	EV 1[1]	MODEL (FULL)[2]	MODEL (RED.)[3]
100/50	**Main**	0.06 (0.02)	0.08 (0.02)	0.08 (0.02)	0.21 (0.03)	----
	only	0.16 (0.03)	0.17 (0.03)	0.17 (0.03)	0.40 (0.04)	----
	SxS	0.04 (0.02)	0.04 (0.02)	0.08 (0.02)	0.23 (0.03)	0.16 (0.03)
	Int.	0.03(0.02)	0.01 (0.01)	0.24 (0.03)	0.39 (0.04)	0.32 (0.04)
1000/500	**Main**	0.05 (0.02)	0.08 (0.02)	0.10 (0.04)	0.23 (0.03)	----
	only	0.15 (0.03)	0.10 (0.03)	0.18 (0.03)	0.48 (0.03)	----
	SxS	0.04 (0.02)	0.05 (0.02)	0.04 (0.02)	0.21 (0.03)	0.13 (0.03)
	Int.	0.03 (0.02)	0.07 (0.02)	0.18 (0.03)	0.39 (0.04)	0.26 (0.03)

The effect sizes for each of the functional variables are shown in Table 1. The adjusted r-squared values range from 0.21-0.48 for the full model, representing values effect sizes have been seen in other studies [8] [10] [11].

Gene expression and outcome data were generated using scripts written for the R statistical software package (R 2.13.0 [12]).

2.2 Data Analysis

ATHENA

ATHENA is a multi-functional software package that uses grammatical evolution to optimize artificial neural networks, ANNs (GENN) or symbolic regression, SR (GESR) as previously described [13] [14]. Specifically, the algorithm for both GENN and GESR is as follows:

1. The data set is divided into 5 equal parts for 5-fold cross-validation (4/5 for training and 1/5 for testing).
2. Training begins by generating a random population of binary strings initialized to be functional ANNs or SRs. The population is divided into demes across a user-defined number of CPUs for parallelization.
3. The ANNs or SRs in the population are evaluated using the training data and the prediction error for each model is recorded. The solutions with the highest fitness are selected for crossover and reproduction, and a new population is generated.
4. Step 3 is repeated for a pre-defined number of generations. Migration of best solutions occurs between CPUs every n-number of generations, as specified by the user.
5. The overall best solution across generations is tested using the remaining 1/5 data fitness is recorded.
6. Steps 2-5 are repeated four more times, each time using a different 4/5 of the data for training and 1/5 for testing. The best model is defined as the model chosen most over all five cross validations. Ties are broken using the fitness metric (r-squared for quantitative outcomes and balanced accuracy for binary outcomes).

For this analysis, we ran GESR and GENN in ATHENA with different parameter settings for each method because GESR is 2x faster than GENN. For GESR, we used an initial population size of 20000 across 20 demes (1000/deme), 400 generations, and migration between demes every 25 generations. For GENN, we used an initial population size of 8000 across 10 demes (800/deme), 200 generations, and migration every 50 generations. For the biological data analysis we used the following GENN parameters: initial population size of 100,000 across 100 demes (1000/deme), 400 generations, and migration every 25 generations

Random Jungle

Random Jungle (RJ) [15] is a faster implementation of Random Forest (RF). RF is a machine learning algorithm that builds either classification or regression trees from the data to predict a categorical or continuous outcome, respectively [16]. Each tree is trained using a bootstrap sample of individuals from the dataset. For each tree node, the attribute, or independent variable, is selected from a subset of all attributes based on how well it reduces an impurity measure. Individuals that are not used to for tree generation ("out-of-bag" individuals) are used to calculate tree prediction error and assign an importance score to each variable based on the effect permutation has on prediction error [17]. Importantly, RF can detect interactions between variables without large main effects.

RJ was implemented specifically to analyze large quantities of data and can be run on multiple CPUs for faster computation time. Parameter settings for the RJ runs are available from the authors upon request. Importance scores were calculated using the Gini index.

Lasso

Lasso is a linear regression variable selection method [18]. Lasso is different from stepwise regression variable selection in that it operates to minimize the sum of squared errors with a tuning parameter that puts a constraint on the absolute value of the variable coefficients. This results in coefficient shrinkage and allows the methodology to incorporate prediction accuracy and parsimony when generating the final regression model [19].

For this study, we used the lars package in R in order to perform Lasso analysis [20]. The algorithm computes the final solution for all values of the tuning parameter. Default settings were used for all Lasso parameters except the value for "effective zero." This parameter determines the absolute value for the coefficient at which the term is dropped from the model. Based on initial analyses using simulated data, we set the value to 0.005 to put more pressure on parsimony.

Biological Data Set

For this analysis, we used a publicly available data set that consisted of genome-wide SNPs, EVs, and a cytotoxicity measurement generated from 171 HapMap lymphoblastoid cell lines (LCLs). Details of this data set have been previously described [8]. Briefly, cytotoxcity of etoposide, a chemotherapeutic agent, was calculated as IC50, or the concentration of drug at which 50% of the cells remain viable. IC50 values were log-transformed for normalization. Next, we adjusted the quantitative outcome in order to account for relatedness and gender by using the residuals from a mixed model regression analysis in genABEL in R [21]. We reduced the initial number of SNPs downloaded from the HapMap website from ~3 million to ~500,000 by filtering with a minor allele frequency threshold of 0.2, a genotyping rate threshold of 0.9, and linkage disequilibrium pruning with a pair-wise r-squared threshold of 0.9. The EVs consisted of ~18,000 transformed and normalized baseline expression levels.

3 Results

3.1 Random Jungle and Lasso

Figure 2 shows the results for the RJ and Lasso analyses. Detection power is defined as the average number of times the functional variables were identified in the top ranking variables for the simulation models with 150 and 1500 variables. We summarize the results by averaging the detection power across the different effect sizes and types within the two variable counts because our goal is to find the method that is optimal across a variety of models, and these parameters are unlikely to be known a priori in biological data. The null data sets were simulated so that no

variables were forcibly correlated with the outcome. We are showing results for the top 10 and top 3 rankings for each analysis. For Lasso, this was determined by the absolute value of the coefficients in the final regression model, which are normalized so that SNP and expression variables can be compared. For RJ, this was determined by the importance value as described in the methods section.

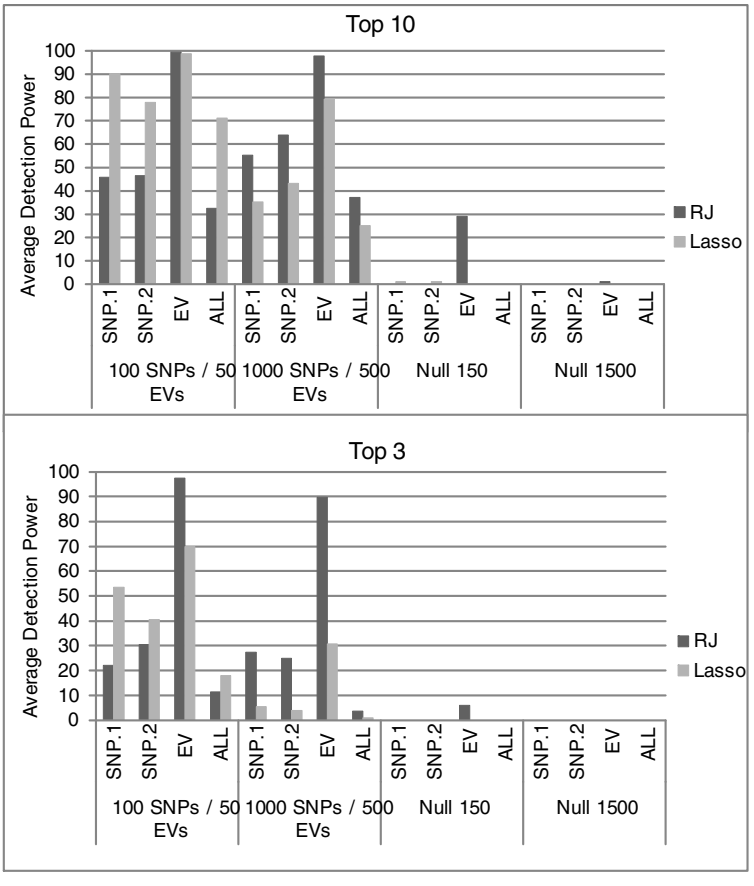


Fig. 2. Average detection power for RJ and Lasso for each of the functional variables (SNP.1, SNP.2, and EV) and the full functional model (ALL) across four different simulated models

Our results show that Lasso had higher average detection for the data sets with 150 variables. Conversely, RJ had higher average detection for the data sets with 1500 variables. Also, RJ appeared to be biased towards ranking EVs higher than SNPs as highlighted by the relatively high detection in of the EV in the null data. Importantly, the Lasso models were not very parsimonious with an average of 23 and 114 variables in the models that had 150 and 1500 total variables, respectively.

3.2 ATHENA

GENN and GESR both generate relatively parsimonious models for all of the analyses. The best models had average variable counts of 4.1 and 2.35 GENN and GESR, respectively). Therefore, we are showing the average number of times the functional variables were identified in the best model (Figure 3). For both variable counts, GENN has higher average detection power than GESR, with the difference most notable in the models with 1500 variables. GENN also performed more accurate modeling than GESR with the average testing set r-squared of 0.25 and 0.15, respectively. (Null data values were -0.001 for GENN and -0.005 for GESR).

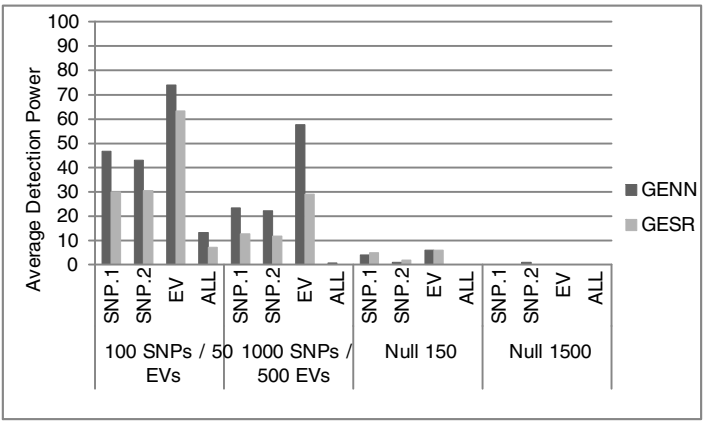


Fig. 3. Average detection power for ATHENA methods GENN and GESR for each of the functional variables (SNP.1, SNP.2, and EV) and the full functional model (ALL) across four different simulated models

3.3 Combination Approach on Biological Data

Based on these results, we developed a RJ filtering-GENN modeling method and applied it to the full biological data set described in the methods section. First, we chose to filter the full data set with RJ because it: 1. can handle large quantities of data in a computationally efficient manner because it is easily parallelized, 2. ranks the variables in a manner that makes filtering simple by using an importance score cut-off or a pre-determined number of variables; and 3. obtained higher overall detection power than the Lasso, another method that could potentially be used as a filter. Next, we chose the GENN algorithm within ATHENA as our modeling technique over GESR and Lasso because it: 1. had higher detection power and modeling accuracy than GESR, and 2. was far more parsimonious than Lasso resulting in smaller, more interpretable models.

First, we ran RJ and filtered the 500 variables with the highest importance values into GENN. Notably, although the simulated data analyses showed RJ to be biased towards ranking EVs higher than SNPs, the biological data set analysis showed the

opposite with 492 SNPs and only 8 EVs in the top 500 spots. This could be due to a number of factors including the overwhelmingly higher number of SNPs than EVs (28x more). For the ATHENA analyses, the best GENN model consisted of two SNPs (rs2375699 and rs1111599) and two EVs (genes TP53I3 and HIST1H4D) and is shown in Figure 4. The testing r-squared value shows that the model explains about 32% of variation in the quantitative outcome. Interestingly, both SNPs were identified as highly significant in the previous study that used this data set [8]. This is encouraging because we are producing similar findings as a study that used a completely different approach.

These results are promising and they begin the process of validating the use of our novel filtering-modeling method for integrating different types of high-throughput data.

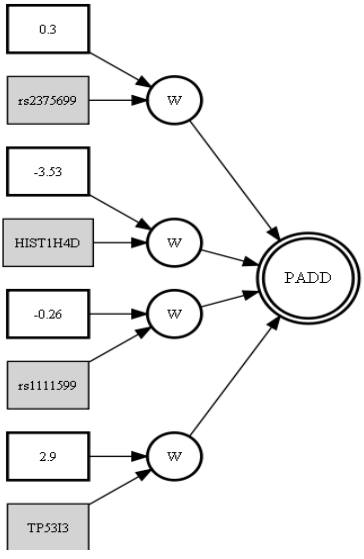


Fig. 4. ANN representing the best GENN model from ATHENA analysis. PADD = additive node; w = multiplication of weight and variable value.

4 Discussion

In order for data analysis to keep up with technology, powerful computational methods must be developed and rigorously tested. For this study, we assessed the ability of three different methods using included simulated genotype and gene expression predictor variables and a quantitative outcome. We then used our results to develop a filtering-modeling technique which plays to the strengths of the different analytical methods. Future studies should involve determining the optimal way to perform RJ filtering. For example, it may be preferable to filter SNPs and EVs separately based on the conflicting biases in simulated and biological data in RJ.

Importantly, the methods tested here are not an exhaustive list of techniques that show potential for meta-dimensional studies. For example, Bayesian networks (BNs) have previously been proposed as a promising method for data integration because they perform variable selection, model both indirect and direct effects, and generate importance values for the variables in the model [22].

The ultimate goal of constructing a powerful meta-dimensional analysis is to determine the biological underpinnings complex human traits. The biological model we generated from our analysis points to potential variants involved in DNA repair; however, follow-up in independent data sets and ultimately functional studies will be needed to determine its implication in etoposide cytotoxicity. The extremely complex, meta-dimensional nature of biology combined with our recent ability to interrogate multiple potential sources of trait variability necessitates novel and creative analysis techniques such as the novel technique implemented in this study.

References

1. Hindorff, L.A., Junkins, H.A., Hall, P.N., Mehta, J.P., Manolio, T.A.: A catalog of published genome-wide association studies (2011)
2. Manolio, T.A., Collins, F.S., Cox, N.J., Goldstein, D.B., Hindorff, L.A., Hunter, D.J., McCarthy, M.I., Ramos, E.M., Cardon, L.R., Chakravarti, A., Cho, J.H., Guttmacher, A.E., Kong, A., Kruglyak, L., Mardis, E., Rotimi, C.N., Slatkin, M., Valle, D., Whittemore, A.S., Boehnke, M., Clark, A.G., Eichler, E.E., Gibson, G., Haines, J.L., Mackay, T.F., McCarroll, S.A., Visscher, P.M.: Finding the missing heritability of complex diseases. Nature 461, 747–753 (2009)
3. Reif, D.M., White, B.C., Moore, J.H.: Integrated analysis of genetic, genomic and proteomic data. Expert. Rev. Proteomics 1, 67–75 (2004)
4. Ideker, T., Dutkowski, J., Hood, L.: Boosting signal-to-noise in complex biology: prior knowledge is power. Cell 144, 860–863 (2011)
5. Chalise, P., Fridley, B.L.: Comparison of Penalty Functions for Sparse Canonical Correlation Analysis. Comput. Stat. Data Anal. 56, 245–254 (2012)
6. Dudek, S.M., Motsinger, A.A., Velez, D.R., Williams, S.M., Ritchie, M.D.: Data simulation software for whole-genome association and other studies in human genetics. Pac. Symp. Biocomput. 11, 499–510 (2006)
7. Edgar, R., Domrachev, M., Lash, A.E.: Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. Nucleic Acids Res. 30, 207–210 (2002)
8. Huang, R.S., Duan, S., Bleibel, W.K., Kistner, E.O., Zhang, W., Clark, T.A., Chen, T.X., Schweitzer, A.C., Blume, J.E., Cox, N.J., Dolan, M.E.: A genome-wide approach to identify genetic variants that contribute to etoposide-induced cytotoxicity. Proc. Natl. Acad. Sci. U S A 104, 9758–9763 (2007)
9. Klein, T.E., Chang, J.T., Cho, M.K., Easton, K.L., Fergerson, R., Hewett, M., Lin, Z., Liu, Y., Liu, S., Oliver, D.E., Rubin, D.L., Shafa, F., Stuart, J.M., Altman, R.B.: Integrating genotype and phenotype information: an overview of the PharmGKB project. Pharmacogenetics Research Network and Knowledge Base. Pharmacogenomics J. 1, 167–170 (2001)
10. Huang, R.S., Duan, S., Shukla, S.J., Kistner, E.O., Clark, T.A., Chen, T.X., Schweitzer, A.C., Blume, J.E., Dolan, M.E.: Identification of genetic variants contributing to cisplatin-induced cytotoxicity by use of a genomewide approach. Am. J. Hum. Genet. 81, 427–437 (2007)
11. Huang, R.S., Duan, S., Kistner, E.O., Bleibel, W.K., Delaney, S.M., Fackenthal, D.L., Das, S., Dolan, M.E.: Genetic variants contributing to daunorubicin-induced cytotoxicity. Cancer Res. 68, 3161–3168 (2008)
12. R Development Core Team: R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria (2011) ISBN: 3900051070, `http://www.R-project.org`
13. Turner, S.D., Dudek, S.M., Ritchie, M.D.: ATHENA: A knowledge-based hybrid backpropagation-grammatical evolution neural network algorithm for discovering epistasis among quantitative trait. Loci. Bio. Data. Min. 3, 5 (2010)
14. Holzinger, E.R., Dudek, S.M., Torstenson, E.C., Ritchie, M.D.: ATHENA Optimization: The Effect of Initial Parameter Settings across Different Genetic Models. In: Giacobini, M. (ed.) EvoBIO 2011. LNCS, vol. 6623, pp. 48–58. Springer, Heidelberg (2011)
15. Schwarz, D.F., Konig, I.R., Ziegler, A.: On safari to Random Jungle: a fast implementation of Random Forests for high-dimensional data. Bioinformatics 26, 1752–1758 (2010)
16. Breiman, L.: Random Forests. Machine Learning 45, 5–32 (2001)

17. Motsinger, A.A., Ritchie, M.D., Reif, D.M.: Novel methods for detecting epistasis in pharmacogenomics studies. Pharmacogenomics 8, 1229–1241 (2007)
18. Tibshirani, R.: Regression Shrinkage and Selection via the Lasso. Journal of the Royal Statistical Society. Series B (Methodological) 58, 267–288 (1996)
19. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least Angle Regression. The Annals of Statistics 32, 407–499 (2004)
20. Hastie, T., Efron, B.: lars: Least Angle Regression, Lasso and Forward Stagewise. R package version 0.9-8 (2011)
21. Aulchenko, Y.S., Ripke, S., Isaacs, A., van Duijn, C.M.: GenABEL: an R library for genome-wide association analysis. Bioinformatics 23, 1294–1296 (2007)
22. Koster, E.S., Rodin, A.S., Raaijmakers, J.A., Maitland-van der Zee, A.H.: Systems biology in pharmacogenomic research: the way to personalized prescribing? Pharmacogenomics 10, 971–981 (2009)

Inferring Phylogenetic Trees Using a Multiobjective Artificial Bee Colony Algorithm

Sergio Santander-Jiménez, Miguel A. Vega-Rodríguez,
Juan A. Gómez-Pulido, and Juan M. Sánchez-Pérez

University of Extremadura,
Department of Technologies of Computers and Communications,
ARCO Research Group,
Escuela Politécnica, Campus Universitario s/n, 10003. Cáceres, Spain
{sesaji,mavega,jangomez,sanperez}@unex.es

Abstract. Phylogenetic Inference is considered as one of the most important research topics in the field of Bioinformatics. A variety of methods based on different optimality measures has been proposed in order to build and evaluate the trees which describe the evolution of species. A major problem that arises with this kind of techniques is the possibility of inferring discordant topologies from a same dataset. Another question to be resolved is how to manage the tree search process. As the space of possible topologies increases exponentially with the number of species in the input dataset, exhaustive methods cannot be applied. In this paper we propose a multiobjective adaptation of a well-known Swarm Intelligence algorithm, the Artificial Bee Colony, to reconstruct phylogenetic trees according to two criteria: maximum parsimony and maximum likelihood. Our approach shows a significant improvement in the quality of the inferred trees compared to other multiobjective proposals.

Keywords: Artificial Bee Colony, Swarm Intelligence, Phylogenetic Inference, Multiobjective Optimization.

1 Introduction

As well as a wide range of topics in Bioinformatics, searching for the best phylogenetic trees which describe the evolution of species is considered as an NP-Hard problem. Several heuristic-based proposals emerged to deal with the computational complexity required by optimality criteria methods like maximum parsimony and maximum likelihood [1]. However, these approaches only consider a single objective to be optimized, so the inference process is carried out in agreement with the chosen criterion. As a result, given a same input dataset, the phylogenies obtained using different methods may be inconsistent with each other. Recent studies have considered the Phylogenetic Inference as an ideal problem to be addressed by multiobjetive optimization techniques [2].

In this paper we try to resolve the Phylogenetic Inference problem according to two metrics: parsimony and likelihood. For this purpose, we define a multiobjective adaptation of the Artificial Bee Colony (ABC), a Swarm Intelligence

M. Giacobini, L. Vanneschi, and W.S. Bush (Eds.): EvoBIO 2012, LNCS 7246, pp. 144–155, 2012.

algorithm based on the colective behaviour of the honey bees [3]. Bioinspired Computing arises as a response to those optimization problems that cannot be solved in reasonable times using classic algorithmic techniques. We have chosen the ABC algorithm because of the promising results which have been reported for a variety of optimization problems [4]. In order to test this algorithm, we have performed a number of experiments on four nucleotide data sets. To illustrate our experimental results, we show the parsimony and likelihood values scored by our Pareto trees and we compared them with the solutions reported by other authors. Additionally, the multiobjective perfomance of our algorithm is evaluated by using the widely used hypervolume metrics.

This paper is organized in the following way. In the next section, we give a brief overview of the published methods for inferring phylogenies. Section 3 explains the basis of Phylogenetic Inference, focusing on the definition of maximum parsimony and maximum likelihood methods. Section 4 describes the proposed algorithm, the Multiobjective Artificial Bee Colony (MOABC). In Section 5 we show, explain and compare the experimental results of our algorithm. Finally, in Section 6 we detail some concluding remarks and define future research lines.

2 Related Work

Phylogenetic Inference has been addressed from different perspectives throughout the years. The first approaches described a number of explicit steps to quickly reconstruct phylogenetic trees from input data. These procedures, known as *algorithmic methods*, do not provide an evaluation function to assess the quality of the generated trees. To overcome this issue, new methods based on optimality measures were proposed. The main goal to achieve with these approaches was the definition of an objective function to be considered in the inference process, generating optimal trees according to some criteria [5]. Some of the most popular *optimality criteria methods* are maximum likelihood and maximum parsimony.

Inferring phylogenetic trees using optimality criteria requires higher processing times than algorithmic methods. Thereby, evolutionary strategies were proposed to deal with this question. The first attempt to apply these ideas was reported by Matsuda in 1995 [6]. Three years later, Lewis developed a genetic algorithm for maximum likelihood inference taking as input nucleotide sequences [7]. This approach laid the foundations for future phylogenetic analyses. Evolutionary algorithms for inferring phylogenetic trees by the maximum parsimony criterion can be also found in the literature [8].

All the previously mentioned approaches suffer from one major drawback: the phylogenetic analyses are performed under a single objective assumption, so the generated topologies may present conflicting ancestral relationships for a same dataset, in agreement with the chosen criterion. Multiobjective optimization emerged as an ideal solution to this [9], allowing new developments to resolve this issue. Coelho et al. published a multiobjective immune-inspired algorithm for inferring phylogenies by minimizing two objective functions: minimal evolution and mean-squared error [10]. In 2007, Cancino and Delbem presented

PhyloMOEA [11], a multiobjective genetic algorithm based on the maximum parsimony and maximum likelihood criteria. The results reported by Cancino and Delbem have motivated this research: a Multiobjective Artificial Bee Colony algorithm for Phylogenetic Inference.

3 Basis of Phylogenetic Inference

Phylogenetic Inference encloses a wide range of estimation techniques that aim to describe ancestral evolutionary relationships among a collection of organisms [5]. These methods take as input a set of n sequences of N characters (often known as sites), which belong to an alphabet α. For example, the alphabet for a DNA-based analysis is $\alpha = \{A, C, G, T\}$, which represents the nucleotide bases Adenine, Cytosine, Guanine and Thymine. These methods use as input molecular characteristics of the organisms under consideration. The output of the inference process is known as *phylogenetic tree*.

A phylogenetic tree is a mathematical structure that represents a hypothesis of the evolution of species. It defines ancestor-descendant relationships among species to explain the data and represents them in a hierarchical tree topology. We can distinguish the following components in a phylogenetic tree:

- Terminal nodes or leaves. They represent the results of the evolutionary history, this is, the input data.
- Internal nodes. They represent hypothetical organisms whose evolution resulted in the species considered as input of the inference process.
- Branches. They indicate an ancestral connection between two nodes. The branches in a tree can be associated to a *branch length* value that usually represents either evolutionary time or molecular changes needed for the evolutionary process.

When a phylogenetic tree has a common ancestor that defines the origin of the phylogeny, we will say that this tree has a *rooted topology*. If this common ancestor does not exist and the direction of natural process cannot be defined, the tree will have an *unrooted topology* [11].

3.1 Methods for Inferring Phylogenies

As we remarked in Section 2, we can find a wide range of methods for inferring phylogenies in the literature. The search for optimal phylogenetic trees is a well-known NP-Hard problem due to the exponential growth of the tree search space in accordance with the number of species. Let n be the number of species to be processed, the number of possible unrooted topologies is given by [12]:

$$\frac{(2n-5)!}{(n-3)!2^{n-3}} \tag{1}$$

This expression means that searching for optimal trees using as input a set of ten or more organisms is cost prohibitive in terms of computational time. The

way to address this problem is the development of evolutionary and bioinspired approaches for inferring optimal topologies in acceptable times.

In the following subsections we will introduce the basis of two of the most used criteria-based methods for phylogenetic reconstruction: maximum parsimony and maximum likelihood analysis.

Maximum Parsimony Approach. Cladistic methods based on the parsimony criterion [1] aim to find those phylogenies that minimize the amount of molecular changes needed to explain the observed data. In a maximum parsimony analysis, the simpler the explanation for natural evolution is, the better the parsimony score will be for that phylogenetic tree. This classic approach is inspired by the Occam's razor principle, which affirms that the simplest explanation for an specific phenomenon will be more plausible than other possible hypotheses.

The parsimony score for a phylogenetic tree τ, inferred from a set of n nucleotide sequences characterized by N aligned sites, is given by the following equation [11]:

$$P(\tau) = \sum_{i=1}^{N} \sum_{(a,b)\in B(\tau)} C(a_i, b_i) \tag{2}$$

where (a, b) is a branch in set B which defines an ancestral relationship between the nodes a and b, a_i and b_i the state or value of the ith site on the sequences for a and b, respectively, and $C(a_i, b_i)$ the cost of evolving from the state a_i to b_i. In a maximum parsimony approach, we will prefer those trees which *minimize* this value, because they would represent a simpler explanation to the observed data. In order to compute the parsimony score, we can find a wide range of proposals in the literature. In this work, we will use the algorithm proposed by Fitch [13] to assess the parsimony of a phylogenetic tree.

Maximum Likelihood Approach. In Phylogenetics, the term likelihood refers to an statistical measure that assesses the probability of the observed data given an evolutionary history described by a tree topology. The main goal in a maximum likelihood approach is the reconstruction of that phylogenetic tree which represents the most likely evolutionary history of the species [1]. In a maximum likelihood analysis, we must bear in mind:

1. The topology of the phylogenetic tree.
2. The branch length values.
3. The molecular evolutionary model.

An evolutionary model, also known as substitution model, describes the probabilities of change from a given state to other one on the molecular sequences of two related organisms. Numerous evolutionary models can be found in the literature (such as JC69, F84, HKY85...) [1]. The likelihood value for a phylogenetic tree will be highly related to the chosen substitution model.

We can formulate the likelihood of a phylogenetic tree as follows. Let τ be the phylogenetic tree to be evaluated, D the set of N-site molecular sequences, D_i the ith state value on the sequences and m the substitution model. The likelihood score is the conditional probability of the data given τ and m [1]:

$$L[D, \tau, m] = \Pr[D|\tau, m] = \prod_{i=1}^{N} \prod_{j=1}^{B} (r_i t_j)^{n_{ij}} \tag{3}$$

where B is the set of branches for τ, r_i the probability of change for the state i, t_j the length of the branch j and n_{ij} the number of state changes in the branch j for the character i. The likelihood is an objective to be *maximized*. The higher the likelihood score for a tree is, the more likely the evolutionary hypothesis will be. We can use the Felsenstein algorithm [14] to compute likelihood.

4 Multiobjective Artificial Bee Colony

In the previous section, we have defined the basis of Phylogenetic Inference and performed a quick review of the methods which define the metrics to be used. Now, we will explain the main features of our proposal, a multiobjective adaptation of the Artificial Bee Colony algorithm for inferring phylogenetic trees.

4.1 Artificial Bee Colony Features

The Artificial Bee Colony is a Swarm Intelligence algorithm proposed by D. Karaboga [3] in 2005. He developed a method to resolve classical optimization problems inspired by the collective behaviour of honey bees. Swarm Intelligence algorithms focus on the definition of a collection of individuals who assume a role in the swarm. These individuals perform their activities and interactuate with others to resolve a problem. This behaviour is governed by well-defined rules and allows the swarm to obtain a collective intelligence. The result is the design of new bioinspired algorithms to address a wide range of problems. Recently, the ABC algorithm has been used to resolve several optimization problems, improving the results of classical evolutionary approaches [4]. This algorithm is inspired by the behaviour of three groups of bees in the hive:

- Employed bees. Employed bees aim to look for and exploit food sources. These bees can examinate the neighbourhood of the current food source they are exploiting and find other new sources.
- Onlooker bees. The information gathered by the employed bees about food sources will be used by onlooker bees to select the most promising sources. These interactions among bees take place in the dancing area. Onlooker bees will decide the sources to be exploited in accordance with the quality of them, denoted by the dances performed by employed bees.
- Scout bees. These bees looks randomly at their environment for new undiscovered food sources. The main purpose of these searches is to avoid the absence of food in the hive when the sources found by other bees are exhausted.

Applying the ABC algorithm to optimization problems, we can identify the bees as the individuals in the population, the food sources as possible solutions to the problem, and the nectar they contain as the fitness of these solutions.

4.2 A Multiobjective Artificial Bee Colony Algorithm for Phylogenetic Inference

In this paper, we propose a multiobjective version of the ABC algorithm applied to Phylogenetic Inference. The main goal is to find those phylogenetic trees which represent a consensus between the maximum parsimony and maximum likelihood criteria. From a multiobjective perspective, these solutions cannot be evaluated in a traditional way because they must simultaneously consider conflicting criteria. To resolve this issue, we apply the *dominance* concept [9]: a solution dominates other one if and only if the first solution has better or equal scores in all considered objectives than the second one and, at least, it is better in one of them. Multiobjective metaheuristics try to obtain those non-dominated solutions which are closer to the optimal solutions to the problem, the set of Pareto-optimal solutions. If we represent Pareto solutions in the value space of n objective functions, the resultant n-dimensional curve is known as Pareto front.

The MOABC algorithm takes as input the following parameters:

1. swarmSize. Population size.
2. maxIterations. Iterations of the main loop to be performed.
3. limit. Control parameter defined to avoid population stagnation.
4. mutation. Mutation rate to be applied over the found solutions to generate new ones.

Our proposal will generate as output multiobjective phylogenetic trees according to the parsimony and likelihood metrics, this is, a set of non-dominated Pareto solutions. Algorithm 1 shows the pseudocode for the MOABC.

The MOABC begins with the initialization of employed bees, which represent the first half of the population. For this purpose, random phylogenetic trees are selected from a repository of 1000 trees generated by bootstrap analysis [1] over each dataset. 500 phylogenetic trees are inferred by maximum parsimony analysis using the DNAPARS software from PHYLIP [15]. The remaining 500 trees are generated by maximum likelihood analysis performed with PhyML [16]. After selecting the initial trees, parsimony and likelihood scores are computed by using the Fitch and Felsenstein algorithms. We use the TreeTemplate class from the C++ libraries for bioinformatics BIO++ [17] to encode phylogenetic trees.

We can differentiate three sections in the MOABC loop. Firstly, employed bees search for solutions in the neighbourhood (lines 6-12 of Algorithm 1). For each employed bee, its associated solution is compared to the result of mutating it. Mutation is carried out by applying Nearest Neighbour Interchange (NNI) topological changes [1] (for parsimony treatment) and modifying randomly selected branch lengths using a gamma distribution [7] (for likelihood treatment),

Algorithm 1. MOABC Pseudocode

```
1: /* Initializing the swarmSize/2 employed bees */
2: C ← initializeAndEvaluatePopulation(swarmSize/2)
3: ParetoFront ← 0
4: i ← 0
5: while i < maxIterations do
6:    for j = 1 to swarmSize/2 do
7:       /* Employed bees: searching for solutions in the neighbourhood*/
8:       newEmployedBee ← generateNeighbour(C[j],mutation)
9:       if MOFitness(newEmployedBee) < MOFitness(C[j]) then
10:         C[j] ← newEmployedBee
11:      end if
12:   end for
13:   /* Generating the probability vector */
14:   probVector ← calculateSelectionProbabilities(C)
15:   /* Generating onlooker bees according to the probability vector */
16:   for j = (swarmSize/2)+1 to swarmSize do
17:      selectedEmployedBee ← selectEmployedBee(probVector,C)
18:      newOnlookerBee ← generateNeighbour(selectedEmployedBee,mutation)
19:      if MOFitness(newOnlookerBee) ≤ MOFitness(selectedEmployedBee) then
20:         C[j] ← newOnlookerBee
21:      else
22:         C[j] ← selectedEmployedBee
23:      end if
24:   end for
25:   /* Generating scout bees */
26:   for j = 1 to swarmSize/2 do
27:      if C[j].iterations > limit then
28:         C[j] ← generateScoutBee()
29:      end if
30:   end for
31:   /* Sorting the current solutions */
32:   C ← FastNonDominatedSort(C)
33:   /* Saving Pareto solutions */
34:   ParetoFront ← saveSolutions(C, ParetoFront)
35:   i ← i + 1
36: end while
```

both according to the *mutation* rate parameter. The NNI operator takes an internal branch of the tree and executes a swap between the nodes in the subtrees situated at the sides of the chosen branch to generate new topologies. In order to improve the likelihood score, we also apply a gradient descent algorithm to optimize tree branch lengths [18].

Once we have generated the neighbour solution, we must decide which one is better in a multiobjective context. For this purpose, Equation 4 is calculated for each competing solution. MOFitness assigns a score to a solution b according to the number of solutions in the population dominated by b and the solutions that dominates b ([19] includes a more detailed explanation). The tree that minimizes this expression will be the solution assigned to the employed bee.

$$MOFitness(b) = Dominates(b) + isDominated(b) * swarmSize \qquad (4)$$

Secondly, onlooker bees (the second half of the population) will decide which solutions must be exploited in accordance with the information provided by employed bees (lines 13-24). For this purpose, current solutions are ordered using two operators taken from NSGAII: *fast non dominated sort* (FNDS) and *crowding distance* [20]. After that, we compute a vector to define selection probabilities

for each solution. The better the solution quality is, the higher its selection probability will be. Onlooker bees will verify this vector and choose one of the current solutions. Neighbour trees are computed by applying mutation and compete with the selected ones using MOFitness. Unlike the previous step, a neighbour solution will be saved if it scores a lower or *equal* MOFitness value with regard to the original solution. Allowing equal scores helps to promote population diversity.

Thirdly, scout bees are generated in the next section (lines 25-30). The *limit* parameter plays a key role in this step. If the solution associated to a bee is not improved in *limit* iterations, it must be discarded (local optimum). This individual becomes a scout bee, which will explore the search space for new solutions. Scout bees randomly select phylogenetic trees from the bootstrap repository and improve them by applying deeper NNI moves and branch length optimization. This strategy allows to avoid local optimal by using different starter trees to explore undiscovered regions of the tree search space.

Once these three sections have been completed, the $swarmSize/2$ best phylogenetic trees found in this iteration are assigned to the employed bees as new starter trees and the MOABC loop begins again. The Pareto set is updated with the best non-dominated solutions and, after *maxIterations*, it will contain those trees whose parsimony and likelihood scores are closer to the optimal values.

5 Experimental Methodology and Results

In this section we explain the methodology to configure our algorithm and show experimental results. Parameters values were assigned in agreement with other authors' proposals [11] with which we will compare our results. For the limit parameter, we performed several experiments to decide its optimal value. For each considered limit value (5, 10, 15, 20 and 25), ten independent runs were carried out and the Pareto sets were evaluated using the hypervolume indicator, a multiobjective metrics that indicates the search space area dominated by our Pareto solutions. The results indicated that the best mean hypervolume values were achieved by using the limit=15 value. Table 1 shows MOABC's configuration.

Our approach was tested on four public nucleotide data sets [11]: $rbcL\_55$ contains 55 sequences (1314 nucleotides per sequence) of the rbcL gene from different species of green plants. $mtDNA\_186$ has 186 sequences (16608 nucleotides per sequence) of human mitochondrial DNA. $RDPII\_218$ is composed of 218 sequences (4182 nucleotides per sequence) of prokaryotic RNA. And $ZILLA\_500$ contains 500 sequences (759 nucleotides per sequence) from rbcL plastid gene.

To prove the statistical relevance of our approach, we have conducted a set of experiments consisting on ten executions per dataset. At the end of them, Pareto fronts were evaluated using the hypervolume metrics and the Shimodaira-Hasewaga (SH) test [21]. Meanwhile hypervolume evaluates solutions from a multiobjective perspective, the SH test decides which percentage of these solutions are not significantly worse than optimal phylogenetic trees found by single-objective approaches [11]. Table 2 resumes our experimental results. For each dataset, Figure 1 shows the Pareto fronts which score the hypervolume value closer to the mean hypervolume obtained by the overall experiments.

Table 1. MOABC input parameters

Parameter	Value
maxIterations	100
swarmSize	100
mutation	5%
limit	15
Topological operator	NNI
Substitution model	HKY85

Table 2. Experimental results

Dataset	Pareto Solutions	Most Parsimonous Tree		Most Likely Tree		SH Test	
		Parsimony	Likelihood	Parsimony	Likelihood	Parsimony	Likelihood
rbcL_55	6	4874	-23969.111	4881	-23961.487	100%	100%
mtDNA_186	14	2431	-40319.181	2448	-40240.283	80%	85%
RDPII_218	34	41488	-149938.023	42621	-144087.843	8%	26%
ZILLA_500	36	16218	-82684.057	16272	-82497.441	98%	61%

Table 3. Hypervolume metrics

Dataset	Minimal Reference Point		Maximum Reference Point		Hypervolume	
					Mean (%)	Std. deviation
	Parsimony	Likelihood	Parsimony	Likelihood		
rbcL_55	4774	-23495.5	5279	-25941.6	64.87	0.0016
mtDNA_186	2376	-39376.4	2656	-43567.2	64.50	0.0005
RDPII_218	40658	-140667.6	45841	-162933.1	70.15	0.0193
ZILLA_500	15893	-80850.9	17588	-89319.8	65.05	0.0007

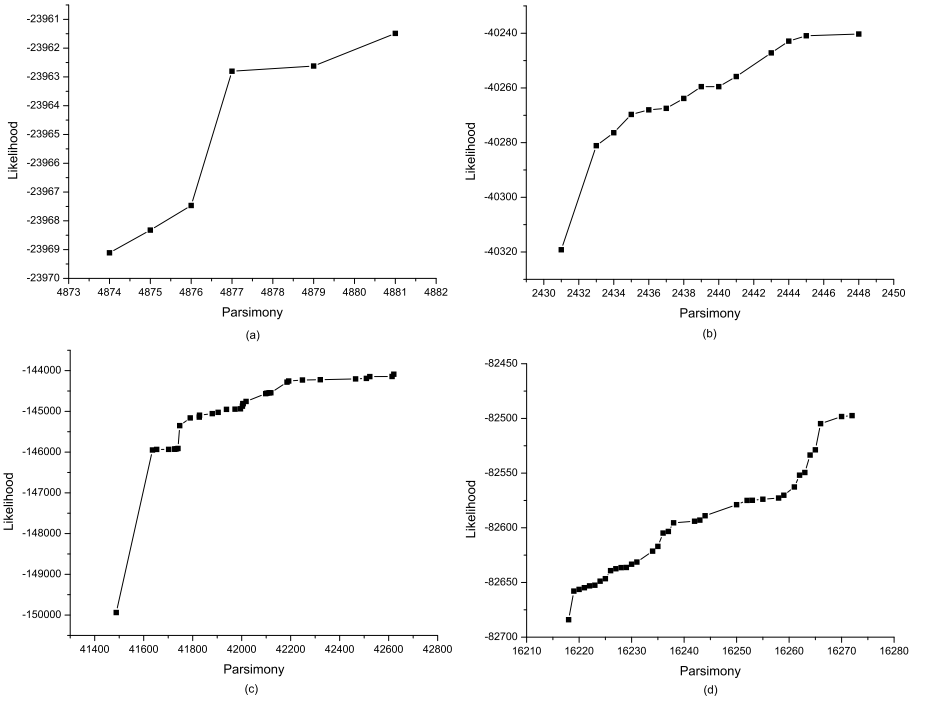


Fig. 1. Pareto fronts for *rbcL_55*(a), *mtDNA_186*(b), *RDPII_218*(c) and *ZILLA_500*(d)

Considering the results for *rbcL_55*, *mtDNA_186* and *ZILLA_500*, the SH test shows a high acceptance ratio because the inferred phylogenetic trees are in

the optimal parsimony-likelihood range. For the $RDPII\_218$ dataset, extreme points on Pareto Front define a higher range of possible non-dominated solutions. This fact motivates a lower acceptance rate. Analyzing these results, we can conclude that the SH test denotes that our extreme solutions are relevant from a single-objective perspective. To evaluate the multiobjective performance of our algorithm, we used the well-known hypervolume indicator. Table 3 defines the reference points and shows mean hypervolume values for each dataset. According to Column 6, our solutions cover over 64% of the space defined by the reference points for $rbcL\_55$, $mtDNA\_186$ and $ZILLA\_500$, and a 70.15% for $RDPII\_218$. We would like to remark that these results are significantly interesting because they would allow researchers to make future comparisons with other bioinspired multiobjective approaches to Phylogenetic Inference.

5.1 Comparisons with Other Authors

In this subsection we present a comparison between our MOABC and PhyloMOEA, a multiobjective algorithm for Phylogenetic Inference published by Cancino and Delbem [11]. Table 4 shows the parsimony and likelihood scores for the most parsimonious and most likely trees (columns 2-3 and 4-5, respectively) found by the two algorithms for each dataset. Our MOABC improves the results reported by Cancino and Delbem in all datasets. For the $rbcL\_55$ instance, the most parsimonious tree found by the two algorithms scores the same parsimony but MOABC improves the likelihood value. This fact demonstrates that our solutions dominate the trees generated by using PhyloMOEA.

In a recent study, Cancino and Delbem suggested the use of a parametric evolutionary model called $HKY85 + \Gamma$ to improve likelihood scores, without changing parsimony values [22]. We have implemented that model in our proposal and carried out new experiments. In Table 5 we can find the likelihood values for the most likely trees found by the two algorithms. Once again, our proposal improves PhyloMOEA's results. Consequently, we conclude that MOABC shows a significant improvement in the quality of the inferred trees.

Table 4. MOABC - PhyloMOEA Comparison

MOABC				
	Best parsimony		Best likelihood	
Dataset	Parsimony	Likelihood	Parsimony	Likelihood
rbcL_55	4874	**-23969.111**	**4881**	**-23961.487**
mtDNA_186	**2431**	**-40319.181**	**2448**	**-40240.283**
RDPII_218	**41488**	**-149938.023**	**42621**	**-144087.843**
ZILLA_500	**16218**	**-82684.057**	**16272**	**-82497.441**
PhyloMOEA				
	Best parsimony		Best likelihood	
Dataset	Parsimony	Likelihood	Parsimony	Likelihood
rbcL_55	4874	-24626.243	4884	-24583.330
mtDNA_186	2438	-41004.302	2450	-40894.343
RDPII_218	41534	-158724.280	42631	-156595.822
ZILLA_500	16219	-87275.281	16276	-86993.825

Table 5. Best likelihood scores using $HKY85 + \Gamma$

Dataset	MOABC	PhyloMOEA
rbcL_55	**-21821.480**	-21889.844
mtDNA_186	**-39890.140**	-39896.441
RDPII_218	**-134149.328**	-134696.535
ZILLA_500	**-80965.400**	-81018.060

6 Conclusions

We have studied in this paper a multiobjective adaptation of a Swarm Intelligence algorithm, the Artificial Bee Colony, to infer phylogenetic trees according to the maximum parsimony and maximum likelihood principles. Our approach has been tested on four public nucleotide data sets and a variety of experiments has been carried out. We have evaluated the multiobjective performance of the proposal by computing the hypervolume metrics and reference points have been defined for future comparisons. Experimental results have proved the relevance of our approach, inferring phylogenetic trees which considerably improve the results reported by other authors in the literature. Therefore, we can suggest that multiobjective Swarm Intelligence algorithms offer multiple possibilities to define improved heuristic approaches to Phylogenetic Inference.

As future research lines, we will address the question of how to boost performance by applying Parallel Computing. Bioinspired techniques and Parallelism will allow us to develop heuristic-based algorithms to infer multiobjective phylogenetic trees minimizing execution times. From this perspective, MPI and OpenMP libraries can help the researcher to exploit the characteristics of modern parallel architectures to improve efficiency. Additional future work could be the study of different topological operators (such as Subtree Pruning and Regrafting (SPR) and Tree Bisection and Reconnection (TBR)) and evolutionary models (JC69, F84...) to optimize parsimony and likelihood scores.

Acknowledgements. This work was partially funded by the Spanish Ministry of Science and Innovation and ERDF (the European Regional Development Fund), under the contract TIN2008-06491-C04-04 (the M* project). We would like to thank the Fundación Valhondo Calaff for the financial support offered to Sergio Santander-Jiménez.

References

1. Felsenstein, J.: Inferring phylogenies. Sinauer Associates, Sunderland (2004); ISBN: 0-87893-177-5
2. Handl, J., Kell, D., Knowles, J.: Multiobjective Optimization in Computational Biology and Bioinformatics. IEEE Transactions on Computational Biology and Bioinformatics 4(2), 289–292 (2006)
3. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Technical report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department (2005)
4. Karaboga, D., Basturk, B.: A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm. Journal of Global Optimization 39(3), 459–471 (2007)
5. Swofford, D., Olsen, G., Waddell, P., Hillis, D.: Phylogenetic Inference. Molecular Systematics, vol. 2, pp. 407–514. Sinauer Associates, Sunderland (1996)
6. Matsuda, H.: Construction of phylogenetic trees from amino acid sequences using a genetic algorithm. In: Proceedings of Genome Informatics Workshop, pp. 19–28. Universal Academy Press (1995)

7. Lewis, P.O.: A Genetic Algorithm for Maximum-Likelihood Phylogeny Inference Using Nucleotide Sequence Data. Molecular Biology and Evolution 15(3), 277–283 (1998)
8. Congdon, C.: GAPHYL: An evolutionary algorithms approach for the study of natural evolution. In: Genetic and Evolutionary Computation Conference, pp. 1057–1064 (2002)
9. Deb, K.: Multi-objective optimization using evolutionary algorithms. Wiley-Interscience Series in Systems and Optimization. John Wiley & Sons, Chichester (2001); ISBN: 978-0-471-87339-6
10. Coelho, G.P., da Silva, A.E.A., Von Zuben, F.J.: Evolving Phylogenetic Trees: A Multiobjective Approach. In: Sagot, M.-F., Walter, M.E.M.T. (eds.) BSB 2007. LNCS (LNBI), vol. 4643, pp. 113–125. Springer, Heidelberg (2007)
11. Cancino, W., Delbem, A.C.B.: A Multi-objective Evolutionary Approach for Phylogenetic Inference. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 428–442. Springer, Heidelberg (2007)
12. Poladian, L., Jermiin, L.: Multi-Objective Evolutionary Algorithms and Phylogenetic Inference with Multiple Data Sets. Soft Computing 10(4), 359–368 (2006)
13. Fitch, W.: Toward Defining the Course of Evolution: Minimum Change for a Specific Tree Topology. Systematic Zoology 20(4), 406–416 (1972)
14. Felsenstein, J.: Evolutionary Trees from DNA Sequences: A Maximum Likelihood Approach. Journal of Molecular Evolution 17, 368–376 (1981)
15. Felsenstein, J.: PHYLIP (Phylogeny Inference Package) (2000), http://evolution.genetics.washington.edu/phylip.html
16. Guindon, S., Dufayard, J.F., Lefort, V., Anisimova, M., Hordijk, W., Gascuel, O.: New Algorithms and Methods to Estimate Maximum-Likelihood Phylogenies: Assessing the Performance of PhyML 3.0. Systematic Biology 59(3), 307–321 (2010)
17. Dutheil, J., Gaillard, S., Bazin, E., Glémin, S., Ranwez, V., Galtier, N., Belkhir, K.: Bio++: a set of C++ libraries for sequence analysis, phylogenetics, molecular evolution and population genetics. BMC Bioinformatics 7, 188 (2006)
18. Press, W., Teukolsky, S., Vetterling, W., Flannery, B.: Numerical Recipes in C, The Art of Scientific Computing. Cambdrige University Press (1992); ISBN: 0–521–43108–5
19. Weicker, N., Szabo, G., Weicker, K., Widmayer, P.: Evolutionary multiobjective optimization for base station transmitter placement with frequency assignment. IEEE Transactions on Evolutionary Computation 7(2), 189–203 (2003)
20. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: NSGA II. IEEE Transactions on Evolutionary Computation 6, 182–197 (2002)
21. Shimodaira, H., Hasegawa, M.: Multiple comparisons of log-likelihoods with applications to phylogenetic inference. Molecular Biology and Evolution 16, 1114–1116 (1999)
22. Cancino, W., Delbem, A.C.B.: A Multi-Criterion Evolutionary Approach Applied to Phylogenetic Reconstruction. In: Korosec, P. (ed.) New Achievements in Evolutionary Computation, pp. 135–156, InTech (2010); ISBN: 978-953-307-053-7

Prediction of Mitochondrial Matrix Protein Structures Based on Feature Selection and Fragment Assembly

Gualberto Asencio-Cortés[1], Jesús S. Aguilar-Ruiz[1], Alfonso E. Márquez-Chamorro[1], Roberto Ruiz[1], and Cosme E. Santiesteban-Toca[2]

[1] School of Engineering, Pablo de Olavide University, Seville, Spain
{guaasecor,aguilar,amarcha,robertoruiz}@upo.es
[2] Centro de Bioplantas, University of Ciego de Ávila, Cuba
cosme@bioplantas.cu

Abstract. Protein structure prediction consists in determining the three-dimensional conformation of a protein based only on its amino acid sequence. This is currently a difficult and significant challenge in structural bioinformatics because these structures are necessary for drug designing. This work proposes a method that reconstructs protein structures from protein fragments assembled according to their physico-chemical similarities, using information extracted from known protein structures. Our prediction system produces distance maps to represent protein structures, which provides more information than contact maps, which are predicted by many proposals in the literature. Most commonly used amino acid physico-chemical properties are hydrophobicity, polarity and charge. In our method, we performed a feature selection on the 544 properties of the AAindex repository, resulting in 16 properties which were used to predictions. We tested our proposal on 74 mitochondrial matrix proteins with a maximum sequence identity of 30% obtained from the Protein Data Bank. We achieved a recall of 0.80 and a precision of 0.79 with an 8-angstrom cut-off and a minimum sequence separation of 7 amino acids. Finally, we compared our system with other relevant proposal on the same benchmark and we achieved a recall improvement of 50.82%. Therefore, for the studied proteins, our method provides a notable improvement in terms of recall.

Keywords: Protein structure prediction, physico-chemical amino acid properties, fragment assembly, protein distance map, feature selection.

1 Introduction

Knowing the protein native 3D structures is currently a difficult and significant challenge because these structures determine protein function and they are necessary to design new drugs. Experimental methods to determine protein structures, generally X-ray crystallography and nuclear magnetic resonance, are very expensive and they have limitations with the structures of some proteins. Moreover,

M. Giacobini, L. Vanneschi, and W.S. Bush (Eds.): EvoBIO 2012, LNCS 7246, pp. 156–167, 2012.

the great number of protein sequences whose three-dimensional structures must be determined, make computational methods of protein structure prediction a very useful tool.

Protein structure prediction (PSP) consists in determining a three-dimensional model based only on the amino acid sequence of a protein and it is currently an issue with great significance in structural bioinformatics [1].

There are currently two main approaches for the PSP problem. The first is the ab initio methods, which find the structure that corresponds to a global minimum of a function, generally a energy function, based in sequence properties. These methods do not use any protein as a template, their computational cost is generally very high and their reliability decreases when the sequence length increases [2].

The second main approach is homology methods, also known as comparative modeling, which try to solve the structure based on protein templates (template-based modeling). This approach is based on the structural conservation of proteins in a protein family, since the 3D structures are more conserved in evolution than sequences. These methods are considered the most currently reliable approach for PSP problem [2].

Template-based modeling methods achieve good results when solved structures are available for proteins with sequences similar to the sequence of the target protein. However, when no homologous proteins with solved structures exist, free-modeling is used.

Within free-modeling methods we find the fragment assembly methods that reconstruct the structure of a protein from structural fragments of other proteins. Three of most relevant fragment assembly-based methods are Fragment-HMM [3], FragFold [4] and ROSETTA [5]. ROSETTA uses a two-stage approach, which begins with a low-resolution model and continues with a representation of all the atoms of the protein, with the goal of minimizing the corresponding energy function.

Since all information used in structure prediction must be inferred from amino acid sequence, there is many useful information derived from sequence used in the literature. Among this information, there are recent methods that use a great set of physico-chemical properties of amino acids [6]. However, the most commonly used properties are hydrophobicity, polarity and charge, which are used, for example, in the models HP and HPNX [7]. There is a database of amino acid properties named AAindex [8] which contains currently 544 properties, from which we selected a subset of 16 in this work by a feature selection process.

The motivation for applying feature selection (FS) techniques has shifted from being optional to becoming a real prerequisite for model building. Specifically, in the PSP problem, the feature selection was also applied and improves the accuracy of predictions [9]. Theoretically, having more features should give us more discriminating power. However, this can cause several problems: increased computational complexity and cost; too many redundant or irrelevant features; and estimation degradation in the classification error.

Based on the generation procedure, FS can be divided into individual feature ranking (FR) and feature subset selection (FSS) [10,11]. FR measures feature-class relevance, then ranks features by their scores and selects the top-ranked ones. These methods are widely used due to their simplicity, scalability, and good empirical success [12]. However, FR is criticized because it can only capture the relevance of features to the target concept, while redundancy and basic interactions between features are not revealed. Furthermore, it is difficult to determine the number of features retained, because a threshold is required. In contrast, FSS attempts to find a set of features that performs well. It integrates the metrics for measuring feature-class relevance and feature-feature interactions.

In this work, a hybrid algorithm was used, BARS [13], in order to handle large datasets to take advantage of the above two approaches (FR, FSS) [14]. This method decouple relevance analysis and redundancy analysis, and have proven to be more effective than ranking methods and more efficient than subset evaluation methods in many traditional high-dimensional datasets.

There are many PSP algorithms currently in the literature that produce a contact map to represent the predicted structure [6,15]. In contrast, our method produces a distance map, which includes more information than a contact map because it incorporates the distances between all of the amino acids in the molecule, irrespective of whether they make contact. There are fewer proposals in the literature that predict distance maps [16], because it is more difficult to perform regression than classification (continuous distances instead binary contacts). Some authors discretize the distances to predict, providing an intermediate representation between contacts and continuous distances, such as the proposal of Walsh et al. 2009 [2] which uses 4-class distance maps. However, unlike 3D models, both distance and contact maps have the desirable property of being insensitive to rotation or translation of the protein molecule.

Our method is a free-modeling approach based on fragment assembly that selects the best distances between pairs of amino acids using fragments of known structures of proteins. These fragments are chosen through a searching process for nearest neighbors by similarity in 16 physico-chemical properties of amino acids selected from the AAindex repository.

We tested our methodology by performing predictions on mitochondrial matrix proteins from the Protein Data Bank (PDB) [17] with a maximum sequence identity of 30%. We have performed predictions with a minimum sequence separation of 7 amino acids, as has been used in the literature [18]. Finally, we compared our system with RBFNN method proposed by Zhang et al. 2005 [19] with the same proteins in the same experimental conditions.

In section 2, we define the elements, procedures and evaluation measures used by our prediction method. In section 3, we detail the used protein datasets, the experimental settings and the achieved results. Finally, in section 4, we describe the main conclusions of the performed study and we outline approaches for future studies.

2 Methods

2.1 Representation of Protein Structures

The representation of protein structure that we used is the distance map, which is a square matrix of order L, where L is the number of amino acids in the protein sequence. The distance matrix is divided in two parts: observed part (upper triangular) and predicted part (lower triangular). The element (i, j), where $i < j$, of the distance matrix is the actual distance measured in angstroms (Å) between the amino acids i^{th} and j^{th} in the sequence. To measure the distances between amino acids, it is necessary to use a reference atom of each amino acid. The most commonly used reference atoms are the alpha carbon and the beta carbon of amino acids [18]. In our method, we used the beta carbon (with the exception of glycine, for which the alpha carbon was used). The distances predicted by the algorithm are stored in the lower triangular of the distance map. Thus, the element (i, j) with $i > j$ of the distance matrix is the predicted distance measured in angstroms between the amino acids i^{th} and j^{th} of the protein sequence.

2.2 Construction of Protein Fragments Knowledge Base

Our prediction system ASPpred (Amino acid Subsequences Properties-based Predictor), works in two phases. In the first phase, it constructs a gallery of protein fragments from all the subsequences of all the proteins in the training set. In the second phase, the target structures of the proteins in test set were predicted using the generated protein fragments model.

The knowledge base consists of a set of vectors called prediction vectors. Each one of these vectors was obtained from one training protein subsequence and contains the physico-chemical properties of the amino acids ends of such fragment. The vector also contains the actual distance between them.

In order to define our prediction vectors formally, it is necessary to define the following elements. In first place, an amino acid sequence of length L is defined by $s_1 \dots s_L$. A fragment or subsequence into a sequence is represented by $s_1 \dots s_b \dots s_e \dots s_L$, where $s_b \dots s_e$ is the fragment, s_b is the beginning amino acid of the fragment, s_e is its ending amino acid and $1 \leq b < e \leq L$.

Moreover, physico-chemical properties are defined by $P_1 \dots P_m$, where m is the number of properties used by the algorithm. The value of the property P_i of an amino acid s_j is defined by $P_i(s_j)$. The prediction vector of a fragment is defined by the tuple showed in Equation 1.

$$\{B_1, E_1, \dots, B_m, E_m, D\} \tag{1}$$

Where D is the distance between amino acids s_b and s_e. B_i and E_i are defined in Equations 2 and 3, respectively. B_i represents the physico-chemical distribution of the entire sequence with decreasing weighting starting at the first amino acid of the fragment. E_i is analogous to B_i starting at the last amino acid of the fragment.

$$B_i = P_i(s_b) + \sum_{\substack{j=1 \\ j \neq b}}^{L} \frac{P_i(s_j)}{L|b-j|}, \forall i \in \{1..m\} \tag{2}$$

$$E_i = P_i(s_e) + \sum_{\substack{j=1 \\ j \neq e}}^{L} \frac{P_i(s_j)}{L|e-j|}, \forall i \in \{1..m\} \tag{3}$$

Note that prediction vectors represent fragments of different lengths, but these lengths is not included in them. The physico-chemical properties included in the prediction vectors are explained in the next subsection. From the point of view of data mining, B_i and E_i are the attributes of training instances and D the class to predict.

2.3 Physico-chemical Feature Selection

To the aim of using the smallest and most effective set of physico-chemical properties, we performed a feature selection from the repository AAindex of physico-chemical properties of amino acids. This repository currently contains 544 amino acid properties.

We used BARS to perform the feature selection over all the properties in AAindex. BARS is an agglomerative algorithm due to the way it constructs the final subset of selected features. The method begins by generating a ranking. Then, pairs of features are obtained with the ranking's first features, in combination with each one of the remaining features on the list. The pairs of features are ranked according to the value of the evaluation, and the process is repeated, that is, the subsets made up by the first sets on the new list are compared with the rest of the sets. At the end, the algorithm returns the best positioned feature subset of all the subsets evaluated.

BARS can use any measure to evaluate feature subsets. Taken into account this domain with a numeric class attribute where distance between amino acids is represented, we used linear regression as evaluator criteria when the search process is carried out to find a relevant and not redundant subset of features.

The dataset that we used for the feature selection is published by Fariselli et al. 2001 [18], that contains 173 proteins with a sequence identity lower than 25%, without chain breaks and with alignments with more than 15 sequences in the corresponding families. This process results on 16 physico-chemical properties that are showed in Table 1 with the same name and description used in AAindex.

2.4 Structure Reconstruction

The second phase of our system consists in obtaining the prediction vectors of the target proteins and in performing a full sequential search to compare each test prediction vector with the training prediction vectors achieved in the first phase. The objective was to find the training prediction vector most similar to

Table 1. The 16 physico-chemical properties of amino acids considered from AAindex

CHOC760104	Proportion of residues 100% buried
LEVM760104	Side chain torsion angle phi(AAAR)
MEIH800103	Average side chain orientation angle
PALJ810107	Normalized frequency of alpha-helix in all-alpha class
QIAN880112	Weights for alpha-helix at the window position of 5
WOLS870101	Principal property value z1
ONEK900101	Delta G values for the peptides extrapolated to 0 M urea
BLAM930101	Alpha helix propensity of position 44 in T4 lysozyme
PARS000101	p-Values of mesophilic proteins based on the distributions of B values
NADH010102	Hydropathy scale based on self-information values in the two-state model (9% accessibility)
SUYM030101	Linker propensity index
WOLR790101	Hydrophobicity index
JACR890101	Weights from the IFH scale
MIYS990103	Optimized relative partition energies - method B
MIYS990104	Optimized relative partition energies - method C
MIYS990105	Optimized relative partition energies - method D

each test prediction vector. For the search process, we consider only training vectors with the same amino acid ends (first and last of each subsequence) than the test vectors. Figure 1 shows this search scheme.

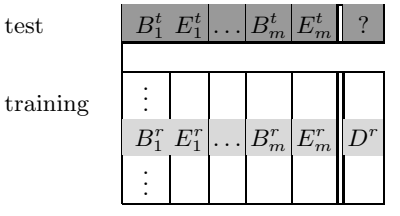


Fig. 1. Search for the most similar training prediction vector

In the search scheme of the Figure 1, $B_1^t \ldots B_m^t$ and $E_1^t \ldots E_m^t$ are the elements of the test subsequence explained above and $B_1^r \ldots B_m^r$ and $E_1^r \ldots E_m^r$ are those of the training subsequence with more similarity to the test subsequence. The distance field D^r of the most similar training vector is assigned to the distance field (symbolized with ?) of test vector.

The training vector with the highest similarity to a test vector satisfies the condition showed in Equation 4. As can be seen in that condition, for the comparison of the test and training vectors, an Euclidean distance is used, which includes all the attributes in these vectors with same weigths. All these attributes are normalised previously. The normalization ensured that all of the attributes are on the same scale and contributed equally to the prediction.

$$\min_{r \in TrainingSet} \sqrt{\sum_{i=1}^{m} (B_i^t - B_i^r)^2 + \sum_{i=1}^{m} (E_i^t - E_i^r)^2} \tag{4}$$

Finally, once predicted distances are assigned in test vectors, these distances are stored in the lower triangular of the distance map of the test sequence. Specifically, the distance field of the prediction vector of the subsequence $s_b \dots s_e$ is assigned to the position (e, b) of the distance map. Thus the structure of each target sequence, by its distance map, is reconstructed.

2.5 Evaluation of Predicted Models

We used several measures to evaluate the quality of the predictions. The first measure is the precision, that is the percentage of predicted contacts that are present in the native structure. This measure is largely used in the literature of protein structure prediction, as in the works of Fariselli et al. [18,20]. The second one is the recall, that is the percentage of native contacts that are predicted to be contacts. Recall has also been widely used in other protein prediction methods [19]. Finally, we have obtained measures of accuracy, specificity and Matthews Correlation Coefficient, that may often provide a much more balanced evaluation of the prediction than percentages [21]. The following formulas (5,6,7,8,9) define these five measures.

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{7}$$

$$Specificity = \frac{TN}{TN + FP} \tag{8}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{9}$$

These measures are used to evaluate the quality of a classification: i.e., each predicted value is assigned a value of 0 or 1. Thus, there are four possible outcomes depending on the quality of the predictions: a) both the real and predicted values are 1 (true positive, TP), b) both the real and predicted values are 0 (true negative, TN), c) the real value is 1 and the predicted value is 0 (false negative, FN) and d) the real value is 0 and the predicted value is 1 (false positive, FP). Because in this case, the class to predict is a real value (a distance), to obtain these measures it is necessary to binarise the class using a distance threshold or cut-off.

In this work, we used a cut-off value of 8 angstroms, which is commonly used in the literature [18,20,19]. In the evaluation of the measures, we omitted predictions of amino acid pairs with a minimum separation in the protein sequence of 7 amino acids, as in Fariselli et al. [18].

3 Experimentation and Results

3.1 Prediction of Mitochondrial Matrix Proteins

We performed an experimental validation of our predictor using all mitochondrial matrix proteins (GO ID: 5759) published in the PDB with a maximum identity of 30% (non-homologous proteins) at October 2011 (74 proteins with a maximum length of 1094 amino acids). In Table 2, we show the PDB codes of the proteins used in this study. We classified proteins in three groups of sequence length L ($L \leq 300$, $300 < L \leq 450$ and $L > 450$) in order to show the prediction behavior for each sequence length interval.

Table 2. Mitochondrial matrix proteins used to train and test our predictor

$L \leq 300$	2CW6	1CSH	2DFD	3CMQ	1PJ3	3C5E
1BWY	2GRA	1D2E	2E0A	3EXE	1WDK	3DLX
1EFV	2HDH	1F0Y	2IB8	3GH0	1WLE	3E04
1KKC	2O23	1GKZ	2IZZ	3KGW	1ZMD	3IHJ
1MJ3	2WYT	1HW4	2OAT	7AAT	2FGE	3IKL
1QQ2	3ED7	1I4W	2QB7	$L > 450$	2J6L	3IKM
1R4W	3EMN	1OTH	2QFY	1A4E	2JDI	3MW9
1RHS	3QUW	1RX0	2R2N	1CJC	2UXW	3N9Y
1TG6	3ULL	1W6U	3AF0	1G5H	2WYA	3OEE
1XX4	5CYT	2A1H	3BLX	1HR6	2XIJ	3OU5
1ZD8	$L300-450$	2BFD	3BPT	1OHV	2ZT5	3SPA

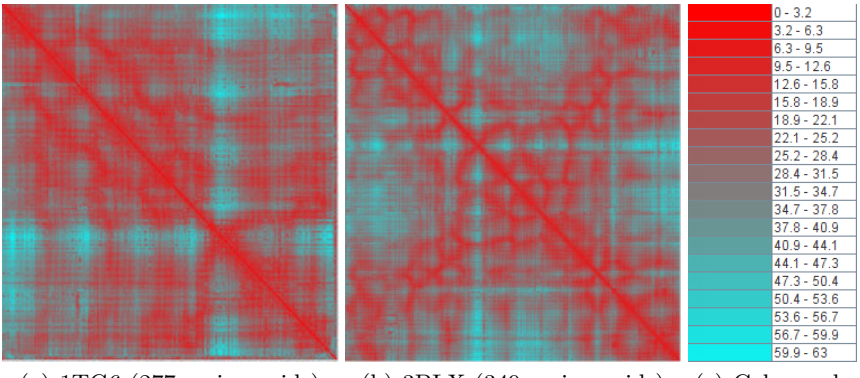
A cross-validation was performed over each group of proteins and over the all 74 proteins. We used a leave-one-out scheme in order to avoid the effect of fold choice in a cross-validation with folds. Table 3 shows the five evaluation measures obtained in this experiment.

We achieved a recall value of 0.80 and a precision of 0.79 for the complete group of proteins, as shown in Table 3. We obtain best predictions, in terms of recall and precision, with proteins of length between 300 and 450 amino acids. In this group of proteins, we achieved recall of 0.84 and precision of 0.83.

In most cases the precision obtained in predicting proteins with long sequences (more than 300 amino acids) is lower than with proteins of short sequences. For example, in the work of Fariselli et al. 2001 [18], which also uses a cross validation, cut-off of 8 angstroms and minimum sequence separation of 7 amino acids, achieved a precision value of 0.11 for proteins of more than 300 amino acids.

Table 3. Efficiency of our method predicting mitochondrial matrix proteins

Protein set	Recall	Precision	Accuracy	Specificity	MCC
All proteins (74)	0.80	0.79	0.97	0.97	0.82
$L \leq 300$ (20)	0.77	0.76	0.98	0.98	0.75
$300 < L \leq 450$ (27)	0.84	0.83	0.99	0.99	0.83
$L > 450$ (27)	0.77	0.76	0.95	0.95	0.82



(a) 1TG6 (277 amino acids) (b) 3BLX (349 amino acids) (c) Color scale

Fig. 2. Predicted distance maps for the mitochondrial matrix proteins 1TG6 (a) and 3BLX (b) with their color scale (c)

Figure 2 shows the predicted distance maps for protein 1TG6 (277 amino acids) and 3BLX (349 amino acids) from the study set. We show a color scale to represent the distances, ranging from the minimum (red) to the maximum (blue) distance. We can appreciate in these distance matrices that the lower triangular (predicted distances) is largely similar to the upper triangular (real distances).

3.2 Comparison with RBFNN on the Same Benchmark

In order to assess the quality of the predictions obtained with our method and to validate our predictor, we compared our proposal with RBFNN method proposed by Zhang et al. 2005 [19]. We predicted the same test proteins with the same training sets in the same conditions.

Zhang et al. 2005 used recall (namely accuracy (A_p) by the authors), predicted and desired numbers to evaluate the performance. Predicted numbers N_p is the count of the predicted contacts by the algorithm and desired numbers N_d is the total number of contacts. The contact threshold was set at 8 Å.

In Table 4 we show the results of this experimentation. As we can see in this table, the average recall (A_p) of ASPpred is 50.82% higher than RBFNN.

Table 4. Comparison at 8 Å with RBFNN on the same benchmark

PDB code (length)	RBFNN			ASPpred		
	N_p	N_d	A_p	N_p	N_d	A_p
1TTF (94)	376	1421	26.46	1307	1421	91.96
1E88 (160)	1006	3352	30.01	3075	3352	91.73
1NAR (290)	3346	10524	31.79	1797	10524	17.07
1BTJ_B (337)	3796	14283	26.58	14026	14283	98.20
1J7E (458)	6589	25026	26.33	23407	25026	93.53
Average			27.67			78.49

N_p: predicted numbers; N_d: desired numbers; A_p: prediction accuracy (%).

Only the protein 1NAR is poorly predicted because there is only one protein as training in the benchmark and it seems to be insufficient to build an effective knowledge base of protein fragments. Thus on the same benchmark dataset, ASPpred yields a sizable improvement.

4 Conclusions and Future Work

In this work we have proposed a method in which protein fragments are assembled according to their physico-chemical similarities, using 16 physico-chemical properties of amino acids selected from AAindex by the BARS feature selection algorithm. We have predicted distance maps, which provide more information about the structure of a protein than contact maps. We have performed an experimental validation of the method on all non-homologous mitochondrial matrix proteins currently available in PDB. We have achieved a recall of 0.80 and a precision of 0.79 with an 8-angstrom cut-off and a minimum sequence separation of 7 amino acids. Finally, we have compared our system with RBFNN method proposed by Zhang et al. 2005 on the same benchmark and we have achieved a recall improvement of 50.82%. Therefore we achieved a significant improvement over previous algorithms.

As future work, we propose to use other prediction vector definitions, including more specific descriptors of the fragment that represent, as amino acid windows. We will also include in these vectors information of the secondary structure of the fragment and its solvent accessibility. We are designing feasibility measures for the geometry derived from predicted distance maps and adjustment algorithms in order to improve our results.

References

1. Zhou, Y., Duan, Y., Yang, Y., Faraggi, E., Lei, H.: Trends in template/fragment-free protein structure prediction. Theoretical Chemistry Accounts: Theory, Computation, and Modeling (Theoretica Chimica Acta) 128, 3–16 (2011)

2. Walsh, I., Bau, D., Martin, A., Mooney, C., Vullo, A., Pollastri, G.: Ab initio and template-based prediction of multi-class distance maps by two-dimensional recursive neural networks. BMC Structural Biology 9(1), 5 (2009)
3. Li, S.C., Bu, D., Xu, J., Li, M.: Fragment-hmm: a new approach to protein structure prediction. Protein Science: A Publication of the Protein Society 17(11), 1925–1934 (2008)
4. Jones, D.T.: Predicting novel protein folds by using fragfold. Proteins (suppl.5), 127–132 (2001)
5. Rohl, C.A., Strauss, C.E.M., Misura, K.M.S., Baker, D.: Protein structure prediction using rosetta. In: Brand, L., Johnson, M.L. (eds.) Numerical Computer Methods, Part D. Methods in Enzymology, vol. 383, pp. 66–93. Academic Press (2004)
6. Li, Y., Fang, Y., Fang, J.: Predicting residue-residue contacts using random forest models. Bioinformatics (2011)
7. Hoque, T., Chetty, M., Sattar, A.: Extended hp model for protein structure prediction. Journal of Computational Biology: A Journal of Computational Molecular Cell Biology 16(1), 85–103 (2009)
8. Kawashima, S., Pokarowski, P., Pokarowska, M., Kolinski, A., Katayama, T., Kanehisa, M.: Aaindex: amino acid index database, progress report 2008. Nucleic Acids Res. 36(Database issue), D202–D205 (2008)
9. Lin, K.-L., Lin, C.-Y., Huang, C.-D., Chang, H.-M., Yang, C.-Y., Lin, C.-T., Tang, C.Y., Hsu, D.F.: Feature selection and combination criteria for improving accuracy in protein structure prediction. IEEE Transactions on NanoBioscience 6(2), 186–196 (2007)
10. Blum, A.L., Langley, P.: Selection of relevant features and examples in machine learning. Artificial Intelligence 97, 245–271 (1997)
11. Guyon, I.: An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157–1182 (2003)
12. Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286, 531–537 (1999)
13. Ruiz, R., Riquelme, J.C., Aguilar-Ruiz, J.S.: Best agglomerative ranked subset for feature selection. Journal of Machine Learning Research - Proceedings Track 4, 148–162 (2008)
14. Yu, L., Liu, H., Guyon, I.: Efficient feature selection via analysis of relevance and redundancy. Journal of Machine Learning Research 5, 1205–1224 (2004)
15. Wu, S., Szilagyi, A., Zhang, Y.: Improving protein structure prediction using multiple sequence-based contact predictions. Structure 19(8), 1182–1191 (2011)
16. Kloczkowski, A., Jernigan, R., Wu, Z., Song, G., Yang, L., Kolinski, A., Pokarowski, P.: Distance matrix-based approach to protein structure prediction. Journal of Structural and Functional Genomics 10, 67–81 (2009)
17. Berman, H., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I., Bourne, P.: The protein data bank. Nucl. Acids Res. 28(1), 235–242 (2000)
18. Fariselli, P., Olmea, O., Valencia, A., Casadio, R.: Prediction of contact maps with neural networks and correlated mutations. Protein Engineering 14(11), 835–843 (2001)

19. Zhang, G.-Z., Huang, D.S., Quan, Z.H.: Combining a binary input encoding scheme with rbfnn for globulin protein inter-residue contact map prediction. Pattern Recogn. Lett. 26, 1543–1553 (2005)
20. Fariselli, P., Casadio, R.: A neural network based predictor of residue contacts in proteins. Protein Engineering 12(1), 15–21 (1999)
21. Baldi, P., Brunak, S., Chauvin, Y., Andersen, C.A.F., Nielsen, H.: Assessing the accuracy of prediction algorithms for classification: an overview. Bioinformatics 16(5), 412–424 (2000)

Feature Selection for Lung Cancer Detection Using SVM Based Recursive Feature Elimination Method

Kesav Kancherla and Srinivas Mukkamala

Institute for Complex Additive Systems and Analysis (ICASA)
Computational Analysis and Network Enterprise Solutuons (CAaNES)
New Mexico Institute of Mining and Technology
Socorro, New Mexico 87801, U.S.A.
{Kancherla,srinivas}@cs.nmt.edu

Abstract. Cancer is the uncontrolled growth of abnormal cells, which do not carry out the functions of normal cells. Lung cancer is the leading cause of death due to cancer in the world. The survival rate of cancer is about 15%. In order to improve the survival rate, we need an early detection method. In this study, we propose a new method for early detection of lung cancer using Tetrakis Carboxy Phenyl Porphine (TCPP) and well-known machine learning techniques. Tetrakis Carboxy Phenyl Porphine (TCPP) is a porphyrin that is able to label cancer cells due to the increased numbers of low density lipoproteins coating the surface of cancer cells and the porous nature of the cancer cell membrane.

In our previous work we studied the performance of well know machine learning techniques in the context of classification accuracy on Biomoda internal study. We used 79 features related to shape, intensity, and texture. We obtained an accuracy of 80% using the current feature set. In order to improve the accuracy of our method, we performed feature selection on these 79 features. We used Support Vector Machine (SVM) based Recursive feature Elimination (RFE) method in our experiments. We obtained an accuracy of 87.5% using reduced 19 feature set.

1 Introduction

Lung cancer is the leading cancer killer among both men and women. Based on statistics by American Cancer Society, it is believed there are 220,000 new cases, deaths per year is about 160,000 and 5-year survival rate of all stages is 15% [1]. However the 5-year survival rate of localized stage is about 50%. Localized stage cancer is the cancer that does not spread to additional sites like lymph nodes within the body. Various factors influencing the 5-year survival rate are stage of cancer, type of cancer, other factors like symptoms, general health etc. Early detection of lung cancer is the leading factor in survival rate. The symptoms of lung cancer do not appear until the cancer spreads to other areas, thus leading to cancer detection of only 24% in early stages [3]. We need an accurate early detection of lung cancer, for increasing the survival rate.

M. Giacobini, L. Vanneschi, and W.S. Bush (Eds.): EvoBIO 2012, LNCS 7246, pp. 168–176, 2012.

Various methods like Computed Tomography (CT) scan, chest radiography, Sputum analysis, microarray data analysis are used for lung cancer detection [5]. Mass screening by Computed Tomography (CT) of chest is promising method for lung cancer detection. However this method is not recommended because it is costly and long term safety of this method is not established due to the risk of exposure to radiation [7]. The use of microarray data for cancer is investigated in [9]. However the use of microarray data is a costly approach. In this paper we investigate the use of Tetrakis Carboxy Phenyl Porphine (TCPP) as alternative approach for early detection of lung cancer.

The use of machine learning for aiding in cancer detection and prediction is investigated in [8]. Machine learning techniques like Artificial Neural Networks (ANN) and Decision Tress (DT) are used for cancer detection for nearly 20 years [10, 11, and 12]. The potential of using machine learning methods for detecting cancer cells or tumors via X-rays, Computed Tomography (CT) is shown in [13, 14]. Machine learning methods used for tumor classification or cancer detection using microarray data or gene expression are Fisher Linear Discriminant analysis [15], K-Nearest Neighbor (KNN) [16], Support Vector Machines (SVM)[17], boosting, and Self-Organizing Maps (SOM) [18], Hierarchical clustering [19], and Graph theoretic approaches [20].

In our previous work [24] we used various machine learning methods for cancer detection using 79 different features. We performed feature selection on this feature set and improved the accuracy to 87.5% using reduced feature set of 19 features. The rest of the paper is organized as follows: in section 2 we describe the data collection process, Tetrakis Carboxy Phenyl Porphine (TCPP) staining procedure, section 3 contains image processing steps and description of features extracted, section 4 contains the results obtained and in section 5 we conclude the paper.

2 Sample Collection

The central hypothesis is that TCPP labeled sputum specimens can detect lung cancer. To test this hypothesis, sputum specimens from various subject cohorts were examined with the Biomoda CyPath® Early Lung Cancer Detection Assay. The long-term goal is to establish the Biomoda CyPath® Early Lung Cancer Detection Assay along with machine learning techniques as an effective program for screening and early detection of lung cancer, with a resultant decrease in lung cancer mortality, and long term monitoring of patients undergoing therapy. A diagnostic and screening tool for lung cancer is important considering that early detection increases survivability. Biomoda CyPath® Early Lung Cancer Detection Assay along with machine learning techniques may provide a useful diagnostic and screening tool method for the early detection of lung cancer and for rapid assessment of the efficacy of tumor therapy and recurrence of lung cancer.

Biomoda's internal study included 28 samples from a variety of sources. Biomoda performed this in-house validation study using sputum samples from 15 lung cancer patients and 13 normal patients. Cohort 1 consisted of 15 patients who had recently been diagnosed with lung cancer and had not undergone surgery or received adjuvant therapy for lung cancer. Cohort 2 included 13 subjects who were heavy smokers but did not have a history or diagnosis of lung cancer.

This study was initiated with an approved protocol and a copy of the informed consent document that was reviewed and approved by a duly-constituted Institutional Review Board (IRB). Subjects aged 18 and above were included in the study. Patients with a history of angina after minimal exertion, severe obstructive lung diseases (Predicted Forced Expiratory Volume in 1 Second (FEV1)<20% of predicted), uncontrolled asthma (defined as a hospitalization or emergency room visit within the last year, > 2 nocturnal Morning dip index (MDI) uses per month, or daily wheezing), and those on supplemental oxygen or resting Saturation of Peripheral Oxygen (SpO2)% <90 % were excluded from the study. The rationale for these exclusion criteria was to avoid any circumstances that could have aggravated their medical condition, considering that some exertion was required for sputum collection (without which the subjects could not have been able to produce adequate quantity of sputum).

A. Sample Collection and Processing

Obtaining the deep lung sample is very important step in successful accomplishment of the assay. The sputum sample was collected over three days following a "triple morning cough procedure". At the Biomoda laboratory, the samples were processed onto a microscope slide, which contained a monolayer of the sputum cells. After preparing the labeling reagents containing TCPP (Biomoda CyPath® Early Detection Lung Cancer Assay), the slide was immersed in the labeling solution, rinsed, air-dried and cover-slipped. The completed slide was viewed under an ultraviolet microscope utilizing a FITC filter and was observed for the presence of fluorescing red cells and other cellular metrics.

B. Slide Scoring and Analysis

Slide scoring and analysis was carried out using the "CyPath Slide Scoring Procedure" (conducted with UV light with a FITC filter) as described below.

- The slide was placed on the microscope stage so that the edge of the microscope's 20x objective remained at the edge of the cellular area.
- Each slide was scanned in a methodical pattern from one end of cellular area to the other, slightly overlapping the area that had already been scanned.
- The results were interpreted based on the characteristics of cancer cells, normal cells and necrotic cells.

  Each TCPP labeled cell was photographed using the 20x objective and stored on a file. The results were recorded in the Biomoda laboratory research notebook.

3 Feature Extraction

Image processing techniques are applied to the images to extract features that assist in differentiating lung cancer cells vs. normal cells. One of the discriminators used for differentiating lung cancer cells vs. normal cells is that cancer cells glow bright red when TCPP is added. Sputum samples from patients that are diagnosed with lung cancer and sputum samples from normal patients are used for performing this initial study.

Digital image in most of the cases is considered as a 2-dimensional signal with intensity as amplitude of the signal. Image preprocessing is a multi staged process involving multiple steps like, image segmentation, image transformation, image restoration etc. In our study we use multiple image based features (image segmentation, intensity based, shape factors, wavelet based, seeded region growing, orientation and eccentricity, and nucleus segmentation). Initial feature extraction steps used in this study are described below:

A. Image Segmentation

Image segmentation is a process of partitioning image into multiple regions. Image segmentation involves the following steps

- Obtain the intensity values from the image
- Apply threshold using the average of intensity values
- Remove pixels which are surrounded by fewer pixels
- Fill the gaps in each individual component
- Obtain individual components using connectivity

B. Feature Extraction

The features can be divided into four groups. Intensity based, shape based, wavelet based and nucleus based features. Brief description of each feature is given below

- **Intensity and Color Based Features**
  Intensity based features are extracted from intensity image and all color components (Red, Green and Blue). We extract average intensity, minimum intensity, mode, variance, maximum intensity, skewness, kurtosis, and number of pixels with maximum intensity and minimum intensity. We extract the same for Red, Green and Blue components. We extract a total of 44 intensity and color based features.

- **Shape Based Features**
  The next set of features is extracted to capture the shape properties of the segment.
  1. Size of the segment, which is nothing but the number of pixels occupied by the segment. This is also area of the segment.
  2. Aspect ratio is the ration between the smallest diameter to the largest diameter

$$\text{Aspect Ratio} = \frac{d_{min}}{d_{max}} \tag{2}$$

Where d_{min} is the smallest diameter, d_{max} is the largest diameter.

  3. Orientation of the cell
  4. Circularity

$$\text{Circularity} = \frac{4\Pi A}{P^2} \tag{3}$$

Where A is the area of the segment, which is size of segment and P is the perimeter of the segment

- **Wavelet Based Features**
  Wavelet transform is powerful signal processing tool for analyzing signals. The wavelet representation consists of coarse overall approximation together with detail coefficients that influence the function at various scales. It overcomes the problems of Short Time Fourier Transform (STFT) relating to time and space resolution. Discrete Wavelet Transform (DWT) provides high time resolution and low frequency resolution for high frequencies and high frequency resolution and low time resolution for low frequencies.

  The wavelet transform has excellent energy compaction and de-correlation properties, which can be used to effectively generate compact representations that exploit the structure of data. Wavelets can capture both texture and shape information efficiently. Wavelet transform can be obtained by decomposing simultaneously with high-pass filter h and low-pass filter g. The outputs will be detail coefficients (from the high-pass filter) and approximation coefficients (from the low-pass). This decomposition can be repeated further to increase the frequency resolution.

  In our experiments we applied level 3 wavelet decomposition using Daubechies wavelet 'db4'. After applying wavelet transform we will be getting one set of approximate coefficients and three sets of detailed coefficients. We extracted mean, variance, maximum and minimum values from each of these coefficients.

- **Nucleus Based Feature**
  Nucleus based features capture the nucleus properties of the cell. In order to segment nucleus we use Seeded region growing method. Region growing methods [25] are a class of region-based segmentation method which groups pixels or sub regions into a larger regions based on certain criteria. In Seeded Region growing method, we start with a initially set of "seed" points and neighboring pixels which have similar properties (such as gray level, texture, color, shape) that of seed points are added to current set. Initial step in this method is to choose a group of pixels as seed points. After this neighboring pixels are added to this initial if they satisfy properties like difference between intensities is below a certain threshold. After performing nucleus segmentation we extract both shape based and intensity based features for nucleus.

4 Experiments

The Support Vector Machine based Recursive Feature Elimination (RFE-SVM) approach [23] is a popular technique for feature selection, especially in the bioinformatics area. Recursive Feature Elimination (RFE) involves training the classifier with respect to feature set F (i.e optimize the parameters for training dataset); calculate the ranking criterion for all the features F; eliminate the feature with least ranking and repeat training process. In the case of RFE-SVM the ranking criterion is the weights magnitude.

After performing image processing techniques we extract the 79 features as explained in previous section. The dataset consists of 119 data points of which 60 data

points are from cancer samples and 59 are normal samples from different patients. Of these data points 66 percent are used for training and remaining 34 percent are used for testing. In order to eliminate over-fitting problem we used 5-fold cross validation during training process. Non-Linear kernel, Radial basis function (RBF) kernel is used in our experiments. Model selection described in [26] is used to find optimal parameters.

Table 1. accuracy obtained by using various features

Number of features used	Accuracy obtained
79	80
74	80
69	82.5
64	82.5
59	85
54	85
49	85
44	85
39	85
34	82.5
29	82.5
24	85
19	87.5

On this dataset we performed Support Vector Machine (SVM) based Recursive feature Elimination (RFE). The features are ranked based on the weight assigned by Support Vector Machine (SVM). After that we remove least significant features in steps of 5 and perform analysis on the remaining features. The accuracies obtained in each step are given in table 1.

Using our initial feature set we obtained a testing accuracy of 80%. The PAP smear, commonly used as a cytological method, has an accuracy of about 62%. Besides obtaining better accuracy, simple method of Biomoda CyPath® adds to its advantages. The PAP test consists of approximately 27 steps and five reagents, some of which are categorized as hazardous materials and also requires higher level of expertise. For 19 features we obtained an accuracy of 87.5%. Further elimination of features resulted in decrease of accuracy. Of these 19 features 9 features belong to intensity based, 3 features belong to wavelet based, 1 feature belongs to shape and remaining are nucleus based (4 shape and 1 intensity). In the 10 intensity based features green component has 4 features; red has 1 feature and remaining 4 features belong to intensity.

We show the performance of our method using Receiver Operating Characteristic (ROC) curves. The Receiver Operating Characteristic (ROC) curves are generated for SVMs by considering the rate at which true positives accumulate versus the rate at which false positives accumulate with each one corresponding, to the vertical axis and the horizontal axis in Figure 1. The point (0, 1) is the perfect classifier, since it classifies all positive and negative cases correctly. Thus an ideal system will initiate by identifying all the positive examples and so the curve will rise to (0, 1)

immediately, having a zero rate of false positives, and then continue along to (1, 1). Figure 1(a) shows the ROC curve obtained using 79 features and figure 1(b) shows the Roc curve obtained using reduced 19 features.

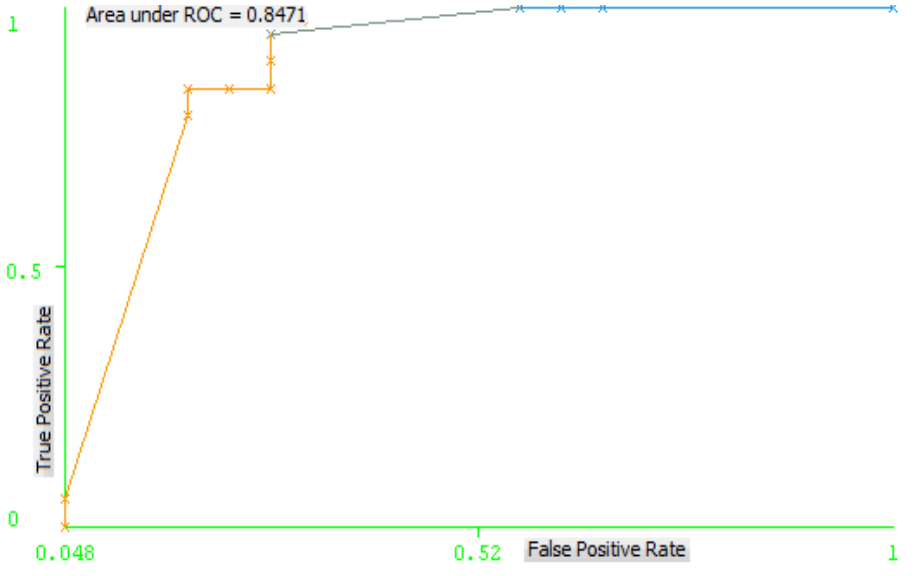


Fig. 1. a ROC curve using 79 features

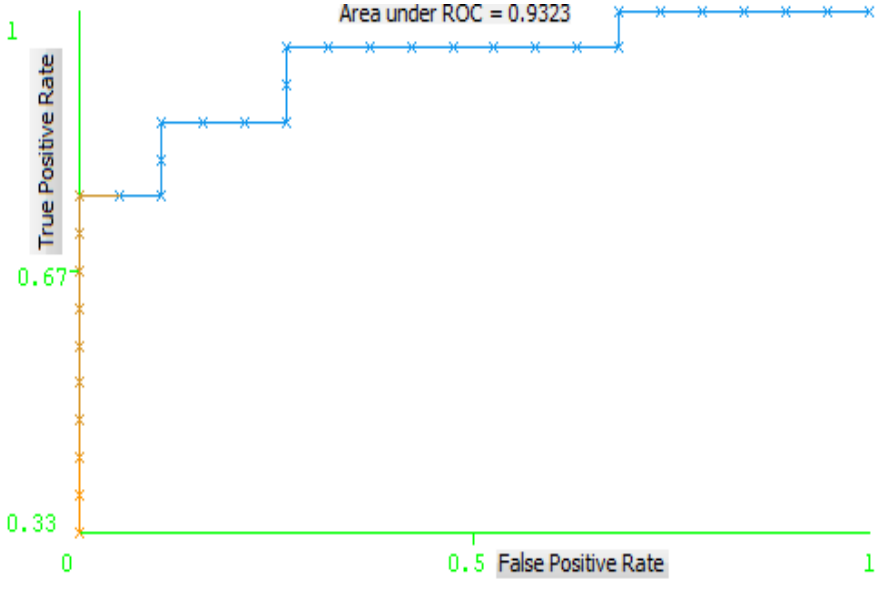


Fig. 1. b ROC curve using 19 features

5 Conclusion

In this paper we performed feature selection using Support Vector Machine (SVM) based Recursive feature Elimination (RFE) method on 79 features. We eliminated features based on their rank and obtained an accuracy of 87.5% by using 19 most significant features. Our experiments show that green component, intensity and shape of the nucleus are significant in detection of lung cancer. Besides its use as a potential screening tool for lung cancer, this method can be used to monitor treatment effectiveness, to detect the recurrence of lung cancer, and also to identify patients who may need an invasive diagnostic procedure. Our results show the potential use of feature selection to improve the accuracy and efficiency of lung cancer detection. As a future work we plan to use large datasets of patients. We would also like to see the performance of other classifiers on this reduced dataset.

References

1. WHO. Deaths by cause, sex and mortality stratum. World Health Organization (2004)
2. Melamed, M.R., Flehinger, B.J., et al.: Screening for Early Lung Cancer. Results of the Memorial Sloan-Kettering Study in New York. Chest 86(1), 44–53 (1984)
3. Fontana, R.S., Sanderson, D.R., et al.: Lung Cancer Screening: the Mayo program. J. Occup. Med. 28(8), 746–750 (1986)
4. Tockman, M.S.: Survival and Mortality from Lung Cancer in a Screened Population: the Johns Hopkins Study. Chest 89(4), S324–S326 (1986)
5. Nanda, K., McCory, P., Myers, E., et al.: Accuracy of the PAP Test in Screening for and Follow up of Cervical Cytologic Abnormalities a Systematic Review. Annals of Internal Medicine,16 132(10), 810–819 (2000)
6. Figge, F.H.J., Weiland, G.S., Manganiello, L.O.J.: Cancer Detection and Therapy. Affinity of Neoplastic, Embryonic and Traumatized Tissues for Porphyrins and Metalloporphyrins. Proc. Soc. Exp. Biol. Med. 68, 640–641 (1948)
7. Taxdal, S.R., Ward, G.E., Figge, F.H.J.: Fluorescence of Human Lymphatic and Cancer Tissues Following High Doses of Intravenous Hematoporphyrin. Surg. Forum. 5, 619–624 (1955)
8. Lipson, R.L., Baldes, E.J., Olsen, A.M.: Hematoporphyrin Derivative: A New Aid for Endoscopic Detection of Malignant Disease. J. Thorac. Cardiovasc. Surg. 42, 623–629 (1961)
9. Galeotti, T., Borrello, S., Palombini, G., Masotti, L., Ferrari, M.B., Cavatorta, P., Arcioni, A., Stremmenos, C., Zannoni, C.: Lipid Peroxidation and Fluidity of Plasma Membranes from Rat Liver and Morris Hepatoma 3924A. FEBS Lett. 169, 169–713 (1984)
10. Galeotti, T., Borrello, S., Minotti, G., Masotti, L.: Membrane Al-terations in Cancer Cells: The Role of Oxy Radicals. Ann. NY Acad. Sci. 488, 468–480 (1986)
11. Campanella, R.: Membrane Lipids Modifications in Human Gliomas of Different Degree of Malignancy. J. Neurosurg. Sci. 36, 11–25 (1992)
12. Liu, H., Kho, A.T., Kohane, I.S., Sun, Y.: Predicting Survival within the Lung Cancer Histopathological Hierarchy Using a Multi-Scale Genomic Model of Development. PLoS Medicine 3(7), 1090–1102 (2006)
13. Ruth, S.V., Baas, P., Zoetmulder, F.A.N.: Surgical Treatment of Malignant Pleural Mesothelioma. Chest Journal 123(2), 551–561 (2003)

14. http://www.mesotheliomahelp.net/default.asp
15. Brown, P., Botstein, D.: Exploring the New World of the Genome with DNA Microarrays. Nature Genetics Supplement 21, 33–37 (1999)
16. Dudoit, S., Fridlyand, J., Speed, T.: Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data. J. Am. Statistical Assoc. 97, 77–87 (2002)
17. Peterson, Ringner, M.: Analysis Tumor Gene Expression Profiles. Artificial Intelligence in Medicine 28(1), 59–74 (2002)
18. Eisen, M., Spellman, P., Brown, P., Botstein, D.: Cluster Analysis and Display of Genome-Wide Expression Patterns. Proc. Nat'l Acad. Sci. USA 95, 14863–14868 (1998)
19. Tamyo, P., et al.: Interpreting Patterns of Gene Expression with Self-Organizing Maps: Methods and Application to Hematopoietic Dif-ferentiation. Proc. Nat'l Acad. Sci. USA 96, 2907–2912 (1999)
20. Liu, D., Fei, S., Hou, Z., Zhang, H., Sun, C.: Advances in Neural Networks, 4th International Symposium on Neural Networks, ISNN 2007 , Nanjing (2007)
21. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Machine Learning 46, 389–422 (2002)
22. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. SIGKDD Exploration 11(1) (2009)
23. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Machine Learning 46, 389–422 (2002)
24. Kancherla, Mukkamala, K., Aveeshaveeshti, S., Cousins, J.: Labeling of Cancer Cells in Sputum for the Early Detection of Lung Cancer Using Tetrakis Carboxy Phenyl Porphine (TCPP). In: IICAI, pp. 1503–1518 (2009)
25. Mancas, M., Gosselin, B., Macq, B.: Segmentation using a region-growing thresholding. In: Proceedings of the SPIE, vol. 5672, pp. 388–398 (2005)
26. Lee, J.H., Lin, C.J.: Automatic model selection for support vector ma-chines. Technical Report, Department of Computer Science and Information Engineering, National Taiwan University (2000)

Measuring Gene Expression Noise in Early *Drosophila* Embryos: The Highly Dynamic Compartmentalized Micro-environment of the Blastoderm Is One of the Main Sources of Noise

Alexander V. Spirov[1,2,*,**], Nina E. Golyandina[3,*,**], David M. Holloway[4], Theodore Alexandrov[5], Ekaterina N. Spirova[6], and Francisco J.P. Lopes[6,***]

[1] Computer Science and CEWIT, SUNY Stony Brook, Stony Brook, New York, USA
[2] The I.M.Sechenov Institute of Evolutionary Physiology & Biochemistry, St.-Petersburg, Russia
[3] Dept. Mathematics and Mechanics, St. Petersburg State University, St.-Petersburg, Russia
[4] Mathematics Dept., British Columbia Institute of Technology, Burnaby, B.C., Canada
[5] Center for Industrial Mathematics, University of Bremen, Germany
[6] AMS, SUNY Stony Brook, Stony Brook, New York, USA
Alexander.Spirov@sunysb.edu, neg@math.spbu.ru

Abstract. Fluorescence imaging has become a widely used technique for quantitatively measuring mRNA or protein expression. The first measurements were on gene expression noise in bacteria and yeast. The relative biological and physicochemical simplicity of these single cells encouraged a number of groups to try similar approaches in multicellular organisms. Such work has been primarily on whole *Drosophila* embryos, where the genes forming the body plan are very well understood. The numerous sources of noise in complex embryonic tissues are a major challenge for characterizing gene expression noise. Here, we present our approach for first separating experimental from biological noise, followed by distinguishing sources of biological noise. We decompose raw signal into trend and residual noise using Singular Spectrum Analysis. We demonstrate our statistical techniques on the *Drosophila* Hunchback protein pattern. We show that the 'texture noise', arising from the pre-cellular compartmentalization of the embryo surface, which is highly dynamic in time, is a major component of total biological noise, and can exceed gene transcription/translation noise.

Keywords: Fluorescence imaging, measuring gene expression, gene expression noise, experimental noise, confocal scanning noise, noise filtration, noise analysis, Singular Value Decomposition (SVD), Singular Spectrum Analysis (SSA), Poisson noise.

---

[*] These authors equally contributed to this work.
[**] Corresponding authors.
[***] Current address: Instituto de Biofisica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil.

M. Giacobini, L. Vanneschi, and W.S. Bush (Eds.): EvoBIO 2012, LNCS 7246, pp. 177–188, 2012.

1 Introduction

The past decade has seen impressive scientific projects aimed at analyzing the noise arising from gene transcription and translation. The approach is based on genetically engineered fluorescent proteins and fluorescence imaging. Specifically, the dual-reporter method was first applied to bacteria [1, 2], and then to yeast cells [3, 4].

These breakthroughs in the study of expression noise at the single cell (and a single gene) level introduced now widely accepted terminology and concepts into the field. The main ideas are that noise in gene expression arises not only from the inherent randomness of biochemical processes such as transcription and translation (intrinsic noise), but also from fluctuations in cellular components (extrinsic noise) that lead indirectly to variation in the expression of a particular gene [e.g.5]. Extrinsic noise is due to cellular components such as regulatory proteins and polymerases, and has a global effect [1]. Intrinsic noise arises from the stochastic nature of the biochemical processes of gene expression and causes identical copies of a gene to express at different levels [1]. These early projects showed that the intrinsic noise in bacteria and yeast tends to be well-fit by a Poisson birth and death process [4; 6].

Detailed consideration of the unicellular experiments raises some crucial questions as to what extent the measured noise is chiefly the molecular noise of gene transcription and translation. Even ignoring the observational noise (e.g. in fluorescence measurements), we still have to pay attention to biological noise sources such as active molecular transport, compartmentalization, the mechanics of cell division, etc. "…one can argue that the inside of a bacterial cell is not a well-stirred pot…" [7]. We might expect that these noise sources are relatively low and controllable in bacterial and yeast populations, but this is certainly not the case for whole embryo observations.

The ideas and approaches for transcription/translation noise in simple single cells has started to be transferred to higher multicellular organisms, primarily the early *Drosophila* embryo [8,9,10,11,12], with tempting parallels between fluorescence of separate nuclei in *Drosophila* blastoderm images and separate cells in the dual-reporter experiments in bacteria and yeast. In both cases we can mask the sources of fluorescence (bacteria / yeast cells and nuclei in the blastoderm), measure their intensity, and estimate noise. The problem is to what extent the nucleus to nucleus variability corresponds to the noise of bacterial gene translation.

Very serious observational limitations to confocal scanning of whole *Drosophila* embryo have been raised by some authors (e.g. Myasnkova et al. [13, 14]). Our experience studying noise in fluorescence intensity in confocal scanning of live gfp-BCD embryos (unpublished) in comparison with fixed immunostained embryos [15] reveals that the nature of the noise is quite far from the simple kinetic considerations of Poisson birth and death processes used in single cells. We think the simple Poisson view does not conform to the biological reality. Nuclear fluorescence, biologically, is the result of complicated and relatively poorly studied processes of active (energy-dependent) transport of diverse molecules from one compartment to another. The geometry and dynamics of the compartments and their relation to the molecular machinery of active transport is still puzzling. The statistics of the nuclear abundance

of a given gene product is not likely to be simply modeled by elementary polypeptide synthesis on the ribosome. We need to pay attention to the many coupled processes of stratification and compartmentalization of the cytoplasm, which is highly dynamic and quite cell-cycle dependent; e.g. with well-known local and whole-embryo scale cytoplasmic movements, especially during mitosis [16].

In this communication, we show that the observed noise of signal intensity in the cytoplasm or in the surface layer of the *Drosophila* blastoderm is largely the result of inhomogeneity of the biological material. We call this 'texture noise'. Fig. 1 illustrates how this texture looks during the syncytial (precellular) blastoderm stage. Texture noise is probably of higher magnitude than the noise arising from the kinetics of protein synthesis (cf Figs. 1&2).

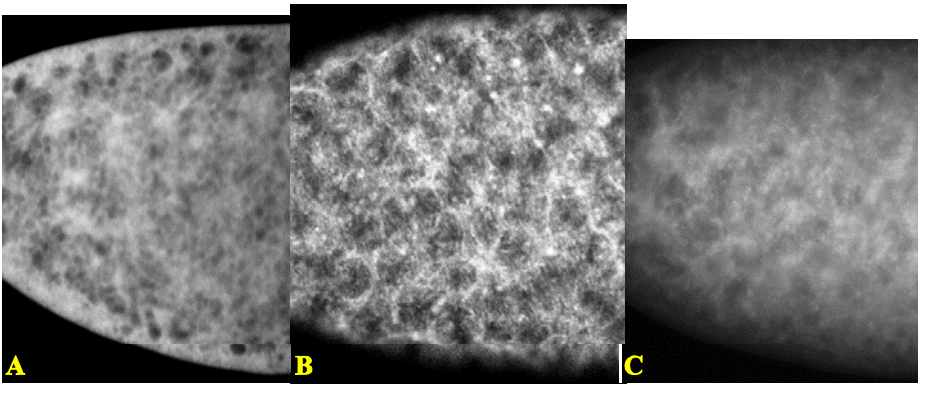


Fig. 1. Some examples of texture in syncytial *Drosophila* embryos. Different techniques show consistent texture characteristics: texture is caused by the geometry of the nuclei (and their close environments), connected by cytoplasmic "bridges". A) protein in a fixed and immnostained embryo; B) live embryo carrying green fluorescence protein; C) mRNA in a fixed embryo, FISH method (fly-FISH DB [17]).

2 Methods and Approaches

2.1 Immunostaining and Confocal Scanning

Embryos were dechorionated, heat fixed, devitellinized, and incubated with primary and secondary antibodies, as described in [22]. Whole-embryo images were taken using a laser confocal scanning microscope (Leica TCS SP2). Images were collected using an HC PL APO 20x objective and variable digital zoom (1.2–1.5x). Each embryo was scanned sequentially, multiple times (64, 128 or 256) and all individual scans were saved. The settings of the microscope were adjusted for each gene product such that pixels expressed at maximum intensity were 200 on the 8-bit scale. Initial image size before processing was 1024x1024 pixels.

2.2 Processing of Sagittal Images and Profile Extraction

We have developed a suite of computational tools to process sagittal images, consisting of a set of plugins for ImageJ software [W.Rasband, NIH USA] and scripts in Delphi (for Wndows OS) or Free Pascal (for Linux OS). After raw image rotation and cropping, the software is used to find the image contour (embryo edges). This contour is then used to find a series of curvilinear profiles running beneath and in parallel with the embryo edge (or contour). Two particular profiles were selected visually: one running through the apical periplasm, the other through the basal periplasm. Local fluorescence intensity is then collected along these profiles, using a small circular window or Region Of Interest (ROI) of given radius R (in pixels), centered on the profile. The ROI is moved down the length of the contour in n-pixel steps. At each step the averaged intensity within the ROI is measured and saved. In this communication we have used a one-pixel ROI (2R=1) and one-pixel step (n=1).

Bleaching Compensation: Our preliminary analysis of the data reveals an approximately linear decrement of intensity with the number of confocal scans. The exact coefficients for the intensity decrease were estimated by linear regression and the data was adjusted on a pixel-to-pixel basis to compensate for fading.

2.3 Singular Spectrum Analysis of Expression Profiles

SSA: Here we describe the basic algorithm for SSA extraction of signal from a one-dimensional series $F = (f_0, \ldots, f_{N-1})$. For the given data, f_n represents the intensity measured at the nth point of a curvilinear profile running through the embryo image. The first step of SSA has only the window length parameter L, $1 < L < N$, and consists of the construction of the Hankel matrix of size $L \times K$, also called the trajectory matrix. There is a one-to-one relation between a series of length N and Hankel matrices of size $L \times K$; each secondary diagonal of a Hankel matrix has equal values and produces a term of the series. The trajectory matrix is then decomposed into the sum of the ordered elementary matrices. This is the so-called singular value decomposition (SVD), and each SVD component generates an elementary reconstructed component (elementary RC) of the series F. The signal extraction problem is thus reduced to (i) choice of window length L and (ii) selection of the subgroup J of SVD components for reconstruction.

Trend Extraction in SSA: SSA needs no a priori specification of models (deterministic or stochastic). In this communication, we are interested in extraction of a slowly varying signal, the trend. Any trend can be approximated by a finite-rank series, as the class of finite-rank series includes all types of sums of products of polynomials, exponentials, and sinusoids. Let us assume that the trend is (or is approximated by) a series of rank r. With large enough N and L ($L \leq N/2$), the trend is separable from noise and is reconstructed by the r-leading SVD components. The subspace spanned by the r-corresponding eigenvectors contains information about the finite-rank structure and, in particular, allows one to derive the approximate analytical formula of the trend. If N is not large enough (e.g. for strong noise or high rank r) and separability is bad, then the trend still can be extracted using small L. In this case, the trend is determined by a few leading components, and SSA works like a smoothing

adaptive linear filter. However, the subspace spanned by the corresponding eigenvectors does not reflect the finite-rank structure of the trend and is liable to be affected by noise and outliers. Grouping of SVD components is based on the fact that the slowly varying component of the series generates eigenvectors and factor vectors of slowly varying form [18], and therefore is composed of similar elementary RCs. Thus, the identification of the components of a trend consists in identification (visual or automatic) of slowly varying eigenvectors, factor vectors, or elementary RCs. Noise is then quantified as the difference between raw intensity and trend value at each position (i.e. local residuals). Fig. 2 illustrates the process of trend extraction on a *Drosophila* gene expression profile. Further details of the SSA application to *Drosophila* data are described in [19; 15].

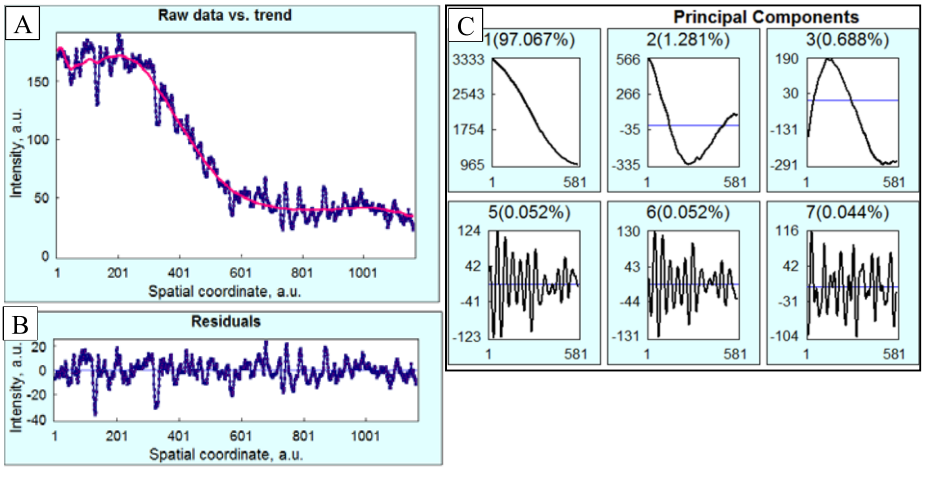


Fig. 2. Decomposition of a raw data series (A, blue) into trend (A, red) and residual noise (B) and criteria for quality of trend extraction (C). The first three principal components are low frequency (C, top), while the rest of components are high frequency and quite different (C, bottom). The sharp transition from low-frequency to high-frequency components indicates a good separation of trend and noise. This data is from the embryo shown in Fig. 3C,D.

Filtering of Periodic Components: With periodic components, extraction is based on Fourier decomposition of the eigenvector elements and assumes the extraction of exponentially-modulated harmonics [18].

2.4 Estimating the Dependence of the Variance on the Trend

To estimate whether the variance of the data depends on the trend, we use the following procedure:

1. adjust for bleaching (see above);
2. calculate means and variances at each pixel over all the scans;
3. Place the (mean, variance) in increasing order;

4. Smooth by calculating a moving average (with the window, say, of 50) of mean/variance.
5. Remove pairs with small mean (less than 20 or 30), as they can be corrupted due to big offset in the microscope.
6. Estimate linear regression coefficients (see section 3.1).

3 Results and Discussion

hunchback (*hb*) is one of the best studied *Drosophila* genes in early embryo development, and one we have worked on for some time [15,20,21,22]. It is a gap-class gene and one of the primary targets of the primary morphogenetic gradient of Bicoid (Bcd). The Bcd-Hb system has been extensively studied with respect to variability and robustness in early embryo patterning [9, 10, 11, 15].

Quantitative analysis of the data, both ours and that presented in the FlyEx database [23] reveal that Hb staining at the beginning of the syncytial blastoderm stage is clearly visible in the cytoplasm (Fig. 3). From nuclear cleavage cycle (cc) 10 to cc 13-14, the cytoplasmic signal decreases several-fold (cf Fig. 4), as Hb is localized to nuclei. I.e., early data is cytoplasmic, and not of nucleus-to-nucleus noise. Visual

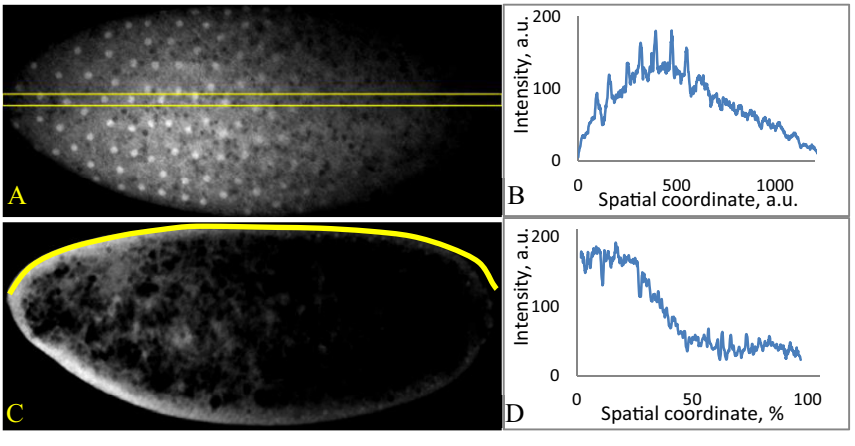


Fig. 3. Hb protein in syncytial cytoplasm showing texture of early embryo. A) Early embryo immunostained for Hb, tangential scan, from cc10, FlyEx database [23]: hz15. B) Intensity plot from a selected strip (about one nucleus wide, marked by yellow borders in A). The seven nuclei clearly seen in the anterior half in A correspond to the seven spikes in B. C) Mid-sagittal scan of an early embryo immunostained for Hb, ~cc 10 - 11. D) The intensity plot acquired by moving an ROI along the yellow profile shown in C.

inspection clearly shows the texture of cytoplasm, and we might expect the residual noise, after elimination of experimental / observational noise, to be dominated by this

cortical texture noise (not transcription-translation noise). The texture noise seen in early cytoplasm is likely to be carried over into older, cellularizing embryos

3.1 Photo-Detection Noise

At the cellularizing blastoderm stage, mid cc 14A, the Hb signal is mainly from nuclei; the signal from the basal cortical cytoplasm is very low but detectable. Here, we compare Hb noise from the nuclear layer versus Hb noise in basal cytoplasm (below the nuclei). We use pixel-width profiles as well as the chains of small ROIs, running through the nuclear layer and through the basal cytoplasmic layer.

A sagittal image of a mid cc 14A embryo immunostained for Hb is shown in Fig. 4A. We present analysis on two selected profiles: the 11th one runs through the brightest area of the nuclear layer, showing the nuclear spatial periodicity; while the 26th one runs far below the nuclear layer has practically no periodic components (see Fig. 4A). Pixel intensity was adjusted as described in section 2, to compensate for bleaching.

Linear Dependence of Pixel Variance on Mean (trend): The embryo in Fig. 4A was scanned 256 times in succession, giving 256 (independent) measurements for each profile. Hence each pixel (point) corresponds to a set of 256 sample values. With this data, we calculate a mean and variance for each pixel (point). Scatterplots for ~1000 datapoints (pixels) are shown for variances (Fig. 4B) and means (Fig. 4C) against anterior-posterior position. Our data show a linear dependence of the variance on the mean (Fig. 4D). The proportionality of the variance to the mean indicates a Poisson distribution; the null hypothesis of a Poisson distribution is not rejected (P-level = 0.4).

This effect can be explained if we are observing a linear combination of a Poisson random variable ξ. If we observe $\eta = a*\xi - b$ (more precisely, $\eta = \max(a*\xi - b, 0)$, where a corresponds to microscope gain, and b corresponds to microscope offset) and $\mathbf{D}\xi = \mathbf{E}\xi = L$ for a Poisson distribution (**D** denotes variance, **E** denotes expectation), then $\mathbf{D}\eta = a^2*L$, $\mathbf{E}\eta = a*L - b$. Therefore, $\mathbf{D}\eta = a*(\mathbf{E}\eta + b) = a*\mathbf{E}\eta + a*b$, where $\mathbf{E}\eta$ (**mean**) and $\mathbf{D}\eta$ (**Var**) are the mean and the variance at each pixel. From the data, we estimate the coefficients in the equation $Var = a*mean + c$ and calculate $b = c/a$.

Results were obtained for several profiles to confirm the reliability of the procedure and the stability of the results. The values of a, b and the ratio a/b was verified by comparing with the real microscope settings for the confocal scans (gain and offset) for the embryos studied. For the Hb data in Fig. 4 the coefficient $a = 1.86$ and $b = 4$. This corresponds to a low offset, which agrees visually with the high background seen in Fig. 4B.

Poisson Character of the Photon-Detection Noise: After the above-described transformation of raw pixel-level data we were able to verify the Poisson character of the instrumental noise. This was done both for individual pixels (analysis of periodograms) and for pixel-width profiles (raw data vs. root-transformed vs. log-transformed ones; data not shown).

We found that observational or confocal scanning noise at the pixel level is dominant in confocal images even after averaging several dozen individual scans. Under the protocols we are using (section 2) it usually takes more than one hundred (up to 256) individual scans of the same embryo to achieve a smooth image. This gradual decrease of pixel noise by averaging over an increasing number of individual scans is evident by eye (data not shown).

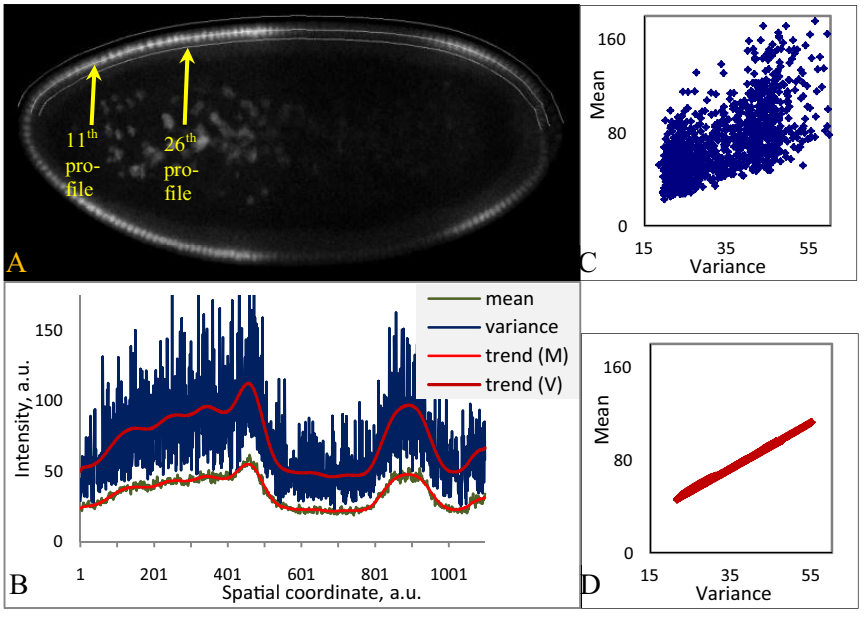


Fig. 4. Observational noise at cc14. A) Sagittal view of an embryo, with the 11th and 26th profles shown. B) Intensities and their variances (red), with their trends, measured along the 11th profile. The variance is proportional to the signal. C) Variance vs. mean (estimated from a 256-size sample) for pixels of the 11th profile: the variance is proportional to the signal. D) Linear dependence of variance on mean (trend).

3.2 Can One Distinguish Photomultiplier Tube (PMT) and Residual Noise?

To elucidate this question we made periodograms for three datasets: from a single scan; and from images made by averaging 32 and 256 scans. In all three cases the right half of the periodogram corresponds mainly to (white) PMT noise. The noise residual after the minimization of the PMT noise corresponds mainly to the left half of the periodogram (from 0 to 0.2) (Fig. 5). Comparing the periodograms (1 scan and 256 scans averaged) we can see that the right component decreases with the number of scans averaged (wide blue arrows). Averaging over a hundred scans leaves mainly the left component of the noise. This noise is not white. We can treat it as a superposition of irregular periodic components mainly belonging to the low frequency half of a periodogram.

Minimization of the PMT noise allows us to study the character of the residual noise, or texture noise as discussed in the Introduction.

For further analysis we concentrated on the 26th profile, the one running underneath the nuclear layer (Fig. 4A) and which does not show any evident sign of nuclear periodicity. Our null hypothesis was that the PMT noise is independent from (or, more strictly, uncorrelated with) the residual (texture) noise. If so, then the variance of total residual noise is a sum of these two components:

$$\mathrm{Var} = \mathrm{Var}_{\mathrm{text}} + \mathrm{Var}_{\mathrm{pmt}}/K,$$

where $\mathrm{Var}_{\mathrm{text}}$ is the variance describing the texture noise, $\mathrm{Var}_{\mathrm{pmt}}$ is the variance describing the PMT noise, and K is the number of scans averaged.

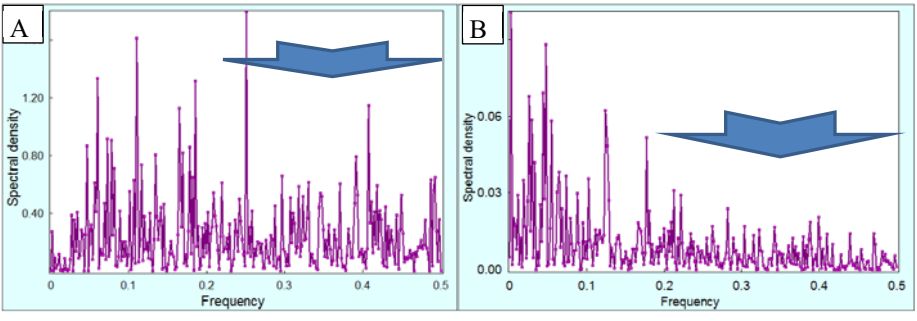


Fig. 5. Decomposition of the residual noise (after subtraction of the trend and the periodic nuclear component) for Hb profile #26 (Fig. 4A) into instrumental (PMT) and residual (texture) noise. The more individual scans used for averaging, the more evident the texture noise components in the periodograms. The texture noise components mainly belong to the frequency interval (0, 0.2), while the PMT noise mainly occupies the frequency interval (0.2, 0.5). (The standard deviation for the PMT noise is 0.36.) Periodograms for the residuals are shown for an individual scan (A) and an average over 256 scans (B).

Four particular data-sets were considered: the 26th profile from a single scan, and from averaging the same profile over 8, 16, and 64 scans. Variance was estimated by SSA, giving the following ratios between $\mathrm{std.dev.}_{\mathrm{text}}$ / $\mathrm{std.dev.}_{\mathrm{pmt}}$:

Scans averaged	$\mathrm{std.dev.}_{\mathrm{text}}$ / $\mathrm{std.dev.}_{\mathrm{pmt}}$
1	0.67 / 2.80
8	0.67 / 0.99
16	0.67 / 0.70
64	0.67 / 0.35

This shows that the pixel-noise observed in images made from averaging 16 scans (the usual practice [24]) is mainly PMT noise, while images made by averaging 64 scans or more is mainly texture noise.

3.3 Texture Noise Character

As shown in previous sections, the trends mined from Hb images averaged for 256 scans include mainly residual noise, which we classify as texture noise. In this section we study the character of the texture noise. A standard way to evaluate the character of the noise in data series is to apply the Box-Cox transformation. A common first step is to compare the results of applying the two extremes of the Box-Cox transformation: square root and logarithm of the data (Fig. 6).

Fig. 6 shows residuals and standard deviations estimated by a sliding window of length = 100 datapoints. For root-transformation, standard deviations are close to constant (Fig. 6D). However, this character of noise-trend dependence is typical for raw (non-transformed) data and sensitive to background removal (data not shown). This raises a more general methodological problem of how to define zero intensity values for experimental data. The zero of confocal scans are set by gain and offset. But the zero of the real signal (measurable quantities of Hb molecules) corresponds to background fluorescence. This background strongly depends on the experimental procedures of fixation and immunostaining, and on antibody qualities. Generally speaking, this subject needs further analysis.

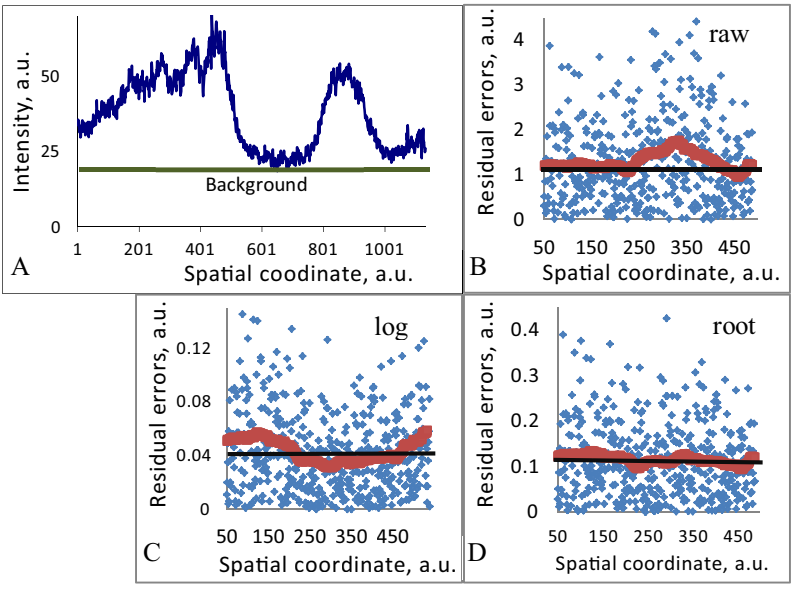


Fig. 6. Poisson character of texture noise for the profile running through the nuclear layer (Hb protein immunostaining; 256 scans, profile 11). A) The raw profile data with background. B) The residual noise (blue) for the profile A after the signal noise decomposition: the noise at higher intensity is higher; the standard deviation estimated by a sliding window of length = 100 datapoints is plotted in red. C) The standard deviation (red), plotted with the residual noise for the log-transformed data. D) The standard deviation (red) plotted with the residual noise for the square-root transformed data.

4 Conclusion

1. The confocal scanning (instrumental) noise at the pixel level is dominant in confocal images. Under common protocols, it takes an averaging of more than one hundred individual scans (of the same embryo) to minimize this noise. We found that this instrumental noise is Poisson distributed.
2. The residual noise after minimization of the instrumental noise appears to be close to Poisson. However the signal-noise dependence is sensitive to the background signal and sensitive to the method of background subtraction.
3. We can treat the residual noise as the superposition of irregular periodic components mainly belonging to the low frequency half of the noise periodogram. Our interpretation is that this noise corresponds to the texture of the biological object we are imaging – the syncytial to cellularizing blastoderm stages of the early *Drosophila* embryo.
4. The residual noise is not likely to be the intrinsic noise of transcription and translation. It is more likely to stem from the spatial compartmentalization of the embryo surface.

Acknowledgements. This work was supported by Joint NSF/NIGMS BioMath Program, 1-R01-GM072022 and the National Institutes of Health, 2R56GM072022-06, 2-R01-GM072022. We thank Center for Developmental Genetics, Stony Brook University, for sharing the confocal microscope.

References

1. Elowitz, M.B., Levine, A.J., Siggia, E.D., Swain, P.S.: Stochastic gene expression in a single cell. Science 297, 1183–1186 (2002)
2. Ozbudak, E.M., Thattai, M., Kurtser, I., Grossman, A.D., van Oudenaarden, A.: Regulation of noise in the expression of a single gene. Nature Genetics 31(1), 69 (2002)
3. Raser, J.M., O'Shea, E.K.: Control of stochasticity in eukaryotic gene expression. Science 304, 1811–1814 (2004)
4. Bar-Even, A., Paulsson, J., Maheshri, N., Carmi, M., O'Shea, E., Pilpel, Y., Barkai, N.: Noise in protein expression scales with natural protein abundance. Nat. Genet. 38, 636–643 (2006)
5. Longo, D., Hasty, J.: Dynamics of single-cell gene expression. Mol. Syst. Biol. 2, 64 (2006)
6. Newman, J.R., Ghaemmaghami, S., Ihmels, J., Breslow, D.K., Noble, M., Derisi, J.L., Weissman, J.S.: Single-cell proteomic analysis of S. cerevisiae reveals the architecture of biological noise. Nature 441, 840–846 (2006)
7. Fedoroff, N., Fontana, W.: Small Numbers of Big Molecules. Science, New Series 297(5584), 1129–1131 (2002)
8. Wu, Y., Myasnikova, E., Reinitz, J.: Master equation simulation analysis of immunostained Bicoid morphogen gradient. BMC Syst. Biol. 1, 52 (2007)
9. Tkacik, G., Gregor, T., Bialek, W.: The Role of Input Noise in Transcriptional Regulation. PLoS ONE 3(7), e2774 (2008)
10. He, F., Wen, Y., Deng, J., Lin, X., Lu, L.J., Jiao, R., Ma, J.: Probing intrinsic properties of a robust morphogen gradient in Drosophila. Dev. Cell 15(4), 558–567 (2008)

11. He, F., Saunders, T.E., Wen, Y., Cheung, D., Jiao, R., ten Wolde, P.R., Howard, M., Ma, J.: Shaping a morphogen gradient for positional precision. Biophys. J. 99(3), 697–707 (2010)
12. Zamparo, L., Perkins, T.J.: Statistical lower bounds on protein copy number from fluorescence expression images. Bioinformatics 25, 2670–2676 (2009)
13. Myasnikova, E., Surkova, S., Panok, L., Samsonova, M., Reinitz, J.: Estimation of errors introduced by confocal imaging into the data on segmentation gene expression in Drosophila. Bioinformatics 25, 346–352 (2009)
14. Myasnikova, E., Surkova, S., Stein, G., Pisarev, A., Samsonova, M.: A regression system for estimation of errors introduced by confocal imaging into gene expression data in situ. BMC Bioinformatics 12, 320 (2011)
15. Holloway, D.M., Lopes, F.J.P., da Fontoura Costa, L., Travençolo, B.A.N., Golyandina, N., Usevich, K., Spirov, A.V.: Gene expression noise in spatial patterning: hunchback promoter structure affects noise amplitude and distribution in Drosophila segmentation. PLoS Comput. Biol. 7(2), e1001069 (2011)
16. Lucchetta, E.M., Vincent, M.E., Ismagilov, R.F.: A Precise Bicoid Gradient Is Nonessential during Cycles 11–13 for Precise Patterning in the Drosophila Blastoderm. PLoS ONE 3(11), e3651 (2008)
17. Lécuyer, E., Yoshida, H., Parthasarathy, N., Alm, C., Babak, T., Cerovina, T., Hughes, T.R., Tomancak, P., Krause, H.M.: Global analysis of mRNA localization reveals a prominent role in organizing cellular architecture and function. Cell 131, 174–187 (2007)
18. Golyandina, N., Nekrutkin, V., Zhigljavsky, A.: Analysis of Time Series Structure: SSA and Related Techniques. Chapman & Hall/CRC, Boca Raton (2001)
19. Alexandrov, T., Golyandina, N., Spirov, A.V.: Singular spectrum analysis of gene expression profiles of early Drosophila embryo: exponential-in-distance patterns. Res. Lett. Signal Processing 2008, 825758 (2008)
20. Lebrecht, D., Foehr, M., Smith, E., Lopes, F.J.P., Vanario-Alonso, C.E., et al.: Bicoid cooperative DNA binding is critical for embryonic patterning in Drosophila. Proc. Natl. Acad. Sci. USA 102, 13176–13181 (2005)
21. Holloway, D.M., Harrison, L.G., Kosman, D., Vanario-Alonso, C.E., Spirov, A.V.: Analysis of pattern precision shows that Drosophila segmentation develops substantial independence from gradients of maternal gene products. Dev. Dyn. 235, 2949–2960 (2006)
22. Lopes, F.J.P., Vieira, F.M.C., Holloway, D.M., Bisch, P.M., Spirov, A.V.: Spatial Bistability Generates hunchback Expression Sharpness in the Drosophila Embryo. PLoS Comput. Biol. 4(9), e1000184 (2008)
23. Pisarev, A., Poustelnikova, E., Samsonova, M., Reinitz, J.: FlyEx, the quantitative atlas on segmentation gene expression at cellular resolution. Nucl. Acids Res. 37, D560–D566 (2009)
24. Surkova, S., Kosman, D., Kozlov, K., Manu Myasnikova, E., Samsonova, A.A., Spirov, A.: Characterization of the Drosophila segment determination morphome. Dev. Biol. 313(2), 844–862 (2008)

Artificial Immune Systems Perform Valuable Work When Detecting Epistasis in Human Genetic Datasets

Delaney Granizo-Mackenzie and Jason H. Moore

Dartmouth College
1 Medical Center Dr.
Hanover, NH 03755, USA
Jason.H.Moore@Dartmouth.edu
http://www.epistasis.org

Abstract. We implement an Artificial Immune System (AIS) for epistasis detection in human genetic datasets. Our AIS outperforms previous attempts to solve the same problem by Penrod et al. by a factor of over 2.4 and performs at 81% of the power of the field standard exhaustive search, Multifactor Dimensionality Reduction (MDR). We show that the immune system performs best when 'paring down' large antibodies to more specific and accurate classifiers. This is promising as it shows that the AIS is doing valuable work, and needs not rely on a near-perfect antibody showing up by chance. We perform a receiver operator characteristic (ROC) analysis to further examine this property.

Keywords: Artificial Immune System, Epistasis, Genetics, Analysis.

1 Introduction

We now have access to vast databases of raw genetic information, but this has in turn made it difficult to determine what part of that information is useful for solving any given problem. The standard approach to determining genetic causes to diseases is to perform a genome-wide analysis study (GWAS) that targets linear genotype-phenotype interactions. However, many are starting to believe that genetic architecture may be more complicated than single-gene, or linear, models; instead it is proposed that complex non-linear interactions can help determine one's phenotypes [6]. These non-linear interactions are termed epistatic interactions and the presence of these interactions is known as epistasis. Epistasis is non-linear because multiple genes may have non-additive effects. In other words, the underlying genetic model for any phenotype may be multiple genes that have no detectable signal individually, but together produce a strong correlation. This view is starting to receive more credence within the genetic community [3].

Performing a linear GWAS is computationally easy. Let G be the set of genes in our dataset, checking each for a correlation takes roughly $O(|G|)$. This, being

M. Giacobini, L. Vanneschi, and W.S. Bush (Eds.): EvoBIO 2012, LNCS 7246, pp. 189–200, 2012.

linear time, is easily achievable on most modern computer systems. However, expanding the search space to include epistatic models means that all combinations of attributes must be considered. This makes the makes the problem combinatorial and non-polynomial. This is prohibitively difficult given the size of the genome and current computing technology. Instead we look for heuristic polynomial-time algorithms that can efficiently detect epistatic signals most of the time; and without resorting to brute-force searches.

The class of algorithms known as artificial immune systems (AISs) takes inspiration from a biological immune system [2]. A biological immune system trains itself to recognize antigens that are not part of the host's body. Similarly, the concept can be adapted to train computer programs to recognize signals that are not part of a healthy genotype. Previous work exists that shows that AISs and related genetic algorithms could be effective in detecting epistatic signals [9,8].

In this paper we demonstrate the performance of our improved AIS in detecting epistatic signals. We also show that the AIS demonstrates some interesting ROC characteristics [4]. Whereas standard ROC's suggests that there should be a trade-off between sensitivity and specificity, experimental analysis shows that maximizing sensitivity produces the most accurate and powerful algorithm. The objective of this work is to demonstrate the implementation of an improved AIS for epistasis analysis compared to previous research and to examine some interesting aspects of the functioning of an AIS.

1.1 Related Work

A basic AIS is This paper builds upon the work done by Penrod et al. [9]. In Penrod et al., the authors implement a basic AIS for detection of epistatic signals in genetic datasets. The paper concluded that whereas the average performance of the AIS did not provide a large improvement over a corresponding random search, there were certain parameter settings that provided better performance. This paper implements an improved AIS, the improvements resulting from an exploration of the speculation presented in Penrod et al. Work on the subject of relating biological, genetic or learning algorithms to epistasis has also been done by Motsinger et al. with GPNN, Greene et al. with Ant Colony Optimization, and Bereta et al. with another Immune System [8,5,1].

2 Methods

2.1 AIS Implementation

We generate antibodies that recognize the input data instances with varying degrees of affinity. Each antibody is an set of rules. The affinity of an antibody for an antigen is calculated by counting the portion of satisfied rules. We then increase the overall fitness of the antibodies over many generations via positive selection and point mutations.

For full source code, please contact the authors at the given email.

Data. We used simulated epistatic datasets generated by the Dartmouth Bioinformatics Laboratory [11]. Each dataset contained $R = 400$ instances and $A = 20$ attributes. Each instance contains the genotype of an individual and an outcome. The genotype consists of a number of single nucleotide polymorphism (SNP) measurements. Each measurement represents the possible zygosity of the corresponding SNP and is represented as either 0, 1 or 2. The zygosity refers to the SNP having two alleles, the possible zygosities are commonly denoted as 'AA', 'Aa' and 'aa'. The outcome classifies the instance as either control or case depending on whether the corresponding phenotype is healthy or sick, respectively. In the simulated data there was no corresponding phenotype, so instances will simply be referred to as 'case' and 'control' abstractly. All testing was done on the same battery of 500 datasets. These were a subset of the datasets used in Penrod et al. Each dataset contained 200 case instances and 200 control individuals. The minor allele frequency was 0.6 and the heritability was 0.01 for all datasets; this is the lowest heritability within all datasets generated by the laboratory, and produces the signals most difficult for algorithms to detect.

Abstract Data Types. For ease of explanation and programming we used several abstract data types (ADTs), namely measurements, functions, rules, antigens and antibodies.

1. **Measurement**: Each attribute in the dataset corresponds to a single nucleotide polymorphism (SNP). Each instance in the dataset contains a measurement of each SNP. We implement measurements as a value that can be either 0, 1 or 2.
2. **Antigen**: Each antigen corresponds to an instance i of the input dataset D. For notational convenience the antigen corresponding to the i^{th} instance is represented as an array of measurements $\mathsf{Ag}_i[]$. $\mathsf{Ag}_i[j]$ refers to the measurement of the j^{th} SNP from the i^{th} instance. $\mathsf{Ag}_i.\mathsf{outcome}$ refers to the outcome of the instances and correspondingly the antigen, it can be either CASE or CONTROL
3. **Function**: A function takes two SNP measurements as input and produces a Boolean output. Specifically

$$f : \{0, 1, 2\} \times \{0, 1, 2\} \rightarrow \{TRUE, FALSE\} \tag{1}$$

   We implemented the equality and inequality functions, or $a = b$ and $a \neq b$ for $a, b \in \{0, 1, 2\}$.
4. **Rule**: A rule contains an index a pointing to an attribute, a function f and a measurement, m. A rule can either be satisfied or violated by an antigen Ag. To determine whether the rule is satisfied we use the index as a pointer to $\mathsf{Ag}[a]$ and apply the rule's function. Specifically

$$R_{a,f,m}(\mathsf{Ag}) = f(\mathsf{Ag}[a], m) \tag{2}$$

5. **Antibody**: Each antibody Ab is a set of rules. The set can be any size from 1 to A, the number of attributes in the dataset and each rule must have a unique index, a.

Terminology and Parameters

1. Case/Control: An antigen is a case antigen if it corresponds to a case instance in the dataset and similarly a control antigen corresponds to a control instance. The set of case antigens is denoted CASES and the set of control antigens is denoted CONTROLS.
2. Wild Card: The number of rules in an antibody can be less than the number of SNPs in an antigen. Therefore for a given antibody there may be some SNPs to which no rule corresponds. For that antibody these orphan SNPs' rules are referred to as wild cards. A wild card indicates that an antibody does not use that SNP as a constraint when predicting outcome.
3. Affinity: This is a measure of how well an antibody 'matches' an antigen. It is equal to the number of satisfied rules over the total number of rules in the antibody for a given antigen.
4. Recognition: If the affinity between an antibody and an antigen is equal to 1.0 we say that the antibody recognizes the antigen.

As parameters for our AIS we implemented:

1. N: Antibody Population Size
2. G: Number of Generations
3. S_r: Survival Rate
4. P_{wc}: Probability of Wild Card
5. M_r: Mutation Rate

Random Antibodies and Mutation. To execute our algorithm we needed two ways of producing new antibodies, namely random generation and mutation.

Random Rule. To return a random rule for attribute a we assign m randomly chosen from Measurements and a function either $=$ or $\neq$ with $p = 1/2$ for both.

Random Antibodies. To generate a random antibody we must produce a set of rules. Our process is not completely random and is biased by our parameter P_{wc}. We start by iterating through each of the n attributes. For each attribute, a, we flip a weighted coin which lands on 'rule' with $p = 1 - P_{wc}$ and 'wild card' with $p = P_{wc}$. If the coin lands on 'rule' we add a random rule with attribute value a to the Ab; else we add nothing, which is equivalent to a wild card.

Mutations. To mutate an antibody we iterate through each of the n attributes. For each attribute, a, we mutate with probability M_r. To mutate attribute a, we flip a weighted coin just as before. If the coin lands on rule we replace the rule referencing attribute a with a random rule; else we eliminate it, which is equivalent to replacing with a wild card.

Calculating Affinity. Antigens and antibodies can be visualized as arrays. Antigens are fixed length and contain attribute measurements of $0, 1$ or 2 along with a case/control classification. The population of antigens is effectively the input dataset parsed into the program. See Table 1, A_i stands for attribute$_i$.

Table 1. Graphical Representation of a set of Random Antigens

Antigen	A_1	A_2	...	A_n	*Status*
1	0	1	...	2	CASE
2	1	1	...	1	CONTROL
...	...	...	...	...	...
A	2	2	...	0	CONTROL

Table 2. Graphical Representation of a set of Random Antibodies

Antibody	R_1	R_2	...	R_{n-1}	R_n
1	$=0$	$*$	...	$=2$	$*$
2	$\neq 1$	$=1$	...	$*$	$*$
...	...	...	...	...	...
N	$*$	$=2$	...	$\neq 0$	$*$

The antibody population is slightly different. To visualize antibodies as arrays we assign rules referencing attribute a to slot a in the antibody array. If no rule exists for attribute a we write a $*$, which represents a wild card. Wild cards are used as stand ins for visualization and do not affect affinity. See Table 2.

To calculate the affinity of an antibody Ab for an antigen Ag we iterate through every rule in Ab and count how many times a rule is satisfied by Ag. Percentage of rules satisfied is termed the affinity, Af of Ab for Ag. Wild cards do not affect the affinity score and are not counted. If the affinity is equal to 1.0 then we say that Ab has recognized Ag. We denote calculating affinity by $\mathsf{Af} = \mathsf{Ab} \cdot \mathsf{Ag}$. This is represented visually in Table 3, rule satisfaction is written as 1 and violation as 0.

Table 3. A Random Antibody Tested Against a Random Antigen

Ag	0	1	0	2	CASE
Ab	$=0$	$*$	$=1$	$\neq 1$	NA
$\mathsf{Ab} \cdot \mathsf{Ag}$	1	N/A	0	1	$\mathsf{Af} = 0.66$

CCRR: We used case to control recognition ratio (CCRR) to score antibodies. To calculate CCRR we take the number of case antigens recognized by an antibody Ab and divide that by the number of control antigens recognized by the antibody. In the event that the number of controls recognized is 0 we add one to the numerator and denominator to avoid division by zero.

Algorithm. We initialize the AIS by generating a population of N random antibodies. We then loop for G generations. In each generation we score the antibodies and then determine an elite selection which will serve as parents. We generate one daughter antibody for each of the parent antibodies by mutating the parent and replacing a non-elite antibody with the mutant. Lastly we replace all remaining non-elite and non-daughter antibodies with new random antibodies. Below we present pseudocode for the algorithm. Within our set of N antibodies we refer to the i^{th} antibody by Ab_i. The affinity of antibody i for antigen j is denoted by $\mathsf{Ab}_i.\mathsf{Af}_j$.

N.B. Any time an integer is required from a real number we round down.

1. INITIALIZATION: Generate A antigens corresponding to the A instances in the input dataset.
2. INITIALIZATION: Generate N random antibodies.
3. For each antibody $\mathsf{Ab}_i | i \in \{1 : N\}$ calculate $\mathsf{Ab}_i.\mathsf{Af}_j = \mathsf{Ab}_i \cdot \mathsf{Ag}_j$ for each control antigen $\mathsf{Ag}_j \in \mathsf{CONTROLS}$.
4. Repeat step 3 for case antigens $\mathsf{Ag}_j \in \mathsf{CASES}$.
5. Set the score of each antibody to its CCRR.
6. Sort antibodies by CCRR in descending order.
7. Replace antibodies Ab_{S_rN+1} through Ab_{2S_rN} with mutations of elite antibodies Ab_1 through Ab_{S_rN} respectively.
8. Replace remaining antibodies Ab_{2S_rN+1} through Ab_N with random antibodies.
9. Repeat steps 3 through 8, G times.
10. OUTPUT: Return the final generation's elite antibodies.

Success Metric. After obtaining the AIS's output of S_rN elite antibodies we counted the number of times that a rule existed for each attribute in the dataset. In other words, we created an array *counts* of size A and then looked at every rule. Every time a rule's attribute was a, we added 1 to $counts[a]$. This served as an indication of how 'important' the AIS judged that attribute to be. We recorded that our algorithm was 'successful' if and only if the first two attributes, $X0$ and $X1$ — the correct genetic model — had the two highest counts within the array.

2.2 Random Search

We observed that our AIS generates $N + G(1 - S_r)N$ new antibodies over the course of execution. To perform a random search corresponding to running an AIS on parameters N, G and S_r we generate NG antibodies, rank them by CCRR and return the NS_r highest scored. We use NG because we can be certain that $NG \geq N + G(1 - S_r)N$ will give the random search at least enough resources. This is equivalent to running the AIS with $N' = NG$ and $G = 1$ while holding all other parameters the same and considering only the top NS_r best scored antibodies for results.

3 Results

3.1 AIS Power

To obtain the performance baseline for our AIS we performed a power analysis over several parameter settings and the 500 dataset testing battery. For each parameter combination we calculated the power by dividing the number of successes by 500. The goal was to determine how well the AIS detects epistatic signals in the most difficult 0.01 heritability datasets. We ran the

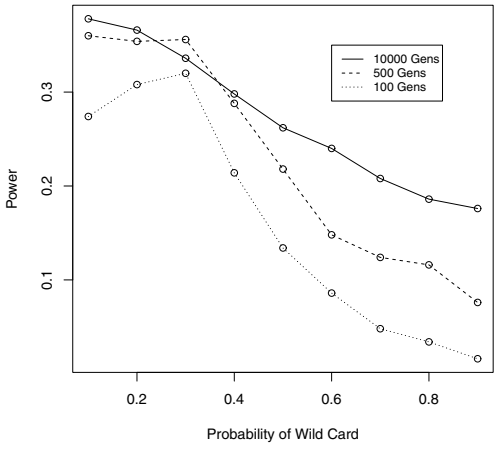


Fig. 1. Graph Showing Average Power of the AIS Varying P_{wc} and G

AIS on the 500 datasets over all combinations of the parameters $N = 100$, $G = \{100, 500, 10000\}$, $S_r = 0.2$, $P_{wc} = \{.1, .2, ..., .9\}$ and $M_r = 0.2$. The results of this analysis are found in Figure 1. We noticed that there were some unexpected powers from the 100 and 500 generation runs for $P_{wc} = .1$ and .2.

We performed a corresponding random analysis. We ran on identical parameters excepting N and G. G was set to 1 and N was set to 10000, 50000, and 1000000 to correspond to the AIS settings of $N = 100$, $G = \{100, 500, 10000\}$. The results of the power analysis are displayed in Figure 2. The random search demonstrated nearly the opposite behavior of AIS. The powers increase as we increase the probability of a wild card.

3.2 MDR Comparison

We tested our AIS against Multifactor Dimensionality Reduction (MDR) — the field standard for epistasis detection [7]. MDR is an exhaustive search and analyzes all possible models of up to m attributes. It therefore takes $O(\frac{A!}{(A-m)!})$ time and is limited by a maximum-way model. The AIS takes approximately $O(G \times N \times A)$ time and is theoretically not bound by a maximum model size. MDR returns a model of each size from 1 to m by generating a full set of IF-THEN rules that encompasses each attribute value combination for the model and testing the consistency of the IF-THEN rules. We ran MDR on the 500 datasets and recorded how many times it returned $X0, X1$ as the two-way model. MDR found the correct result 233 times for a power of 0.466. We then summarized the results of AIS vs. random search vs. MDR in Figure 4. The maximum power of the random search is identical to the minimum power of the AIS at $P_{wc} = .9$. The gap quickly grows as we decrease P_{wc}. The AIS reaches its maximum power of 0.378 at $P_{wc} = .1$, which is 81.12% of MDR's 0.466.

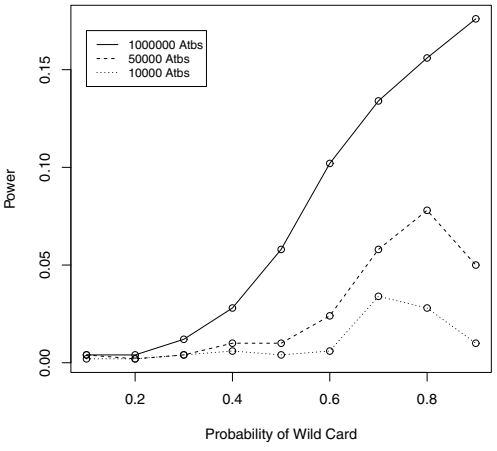


Fig. 2. Graph Showing Average Power of the Random Search Varying P_{wc} and N

Table 4. A power comparison of AIS, random search and MDR

	Powers								
P_{wc}	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Random Search	0.004	0.004	0.012	0.028	0.058	0.102	0.134	0.156	0.176
AIS	0.378	0.366	0.336	0.298	0.262	0.240	0.208	0.186	0.176
MDR	0.466								

3.3 ROC Analysis

If a signal is epistatic, then rules for both of the model attributes $X0$ and $X1$ must be present for an antibody to be scored higher than average. Now if that antibody contains very few rules then it can be considered a very good antibody, as it will be scored very highly and is effectively a silver bullet way of finding the correct model. If the antibody contains many rules, then there will be considerable signal masking by the non-model 'noisy' rules. Therefore we have a trade-off between high-accuracy low-speed and low-accuracy high-speed. This trade-off can be analyzed with ROC principles.

To treat the AIS as a binary classifier systems, we term an antibody as 'correct' if it contains both $X0$ and $X1$ and incorrect otherwise. Through ROC analysis we attempt to determine how the AIS behavior changes as a function of P_{wc}. Since rules are created independently, the chance that a random antibody contains a rule for any attribute is $1 - P_{wc}$. It follows that the chance that the antibody contains any two given attributes is $(1 - P_{wc})^2$. Therefore a random antibody will be correct with a probability of $(1 - P_{wc})^2$.

In an experimental test on the 500 datasets for ($P_{wc} = \{.1, .2, ..., .9\}$, $N = \{100, 10000, 1000000\}$). We found that the actual percentage of correct

antibodies varies from the predicted $(1 - P_{wc})^2$ with a standard deviation of 0.023. For each of the 13500 antibody sets we assigned each antibody a CCRR score based on antigens from the corresponding datasets. Within each set we compared each correct antibody with each incorrect antibody and counted the number of comparisons in which the correct antibody had a higher CCRR. We then divided these counts by the total number of comparisons.[1] Figure 3 shows the results of this analysis.

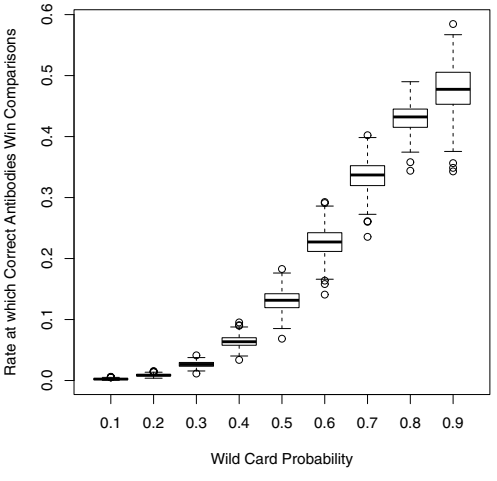


Fig. 3. Plot showing percentage of comparisons won by correct antibodies

As predicted a lower P_{wc} causes a reduction in signal strength.[2] Finally, we multiplied the comparison win rate and the correct antibody generation chance. Figure 4 shows the resulting values. We see that $P_{wc} = .6$ appears to be the best value for facilitating both presence of the correct model and lack of signal-masking.

[1] Statisticians will note that a Wilcoxon test could be used to determine how significantly our scoring metric places correct antibodies above incorrect antibodies [10]. We did not perform a Wilcoxon test as it would have been impractical and would not have lend any information of real value for our purposes. It may be of use to researchers, however, when attempting to determine which scoring metrics are optimal for the AIS algorithm.

[2] The percentage of comparisons won falls below the expected 0.5 in nearly all cases. When the wild card probability is low we have very large and specific antibodies generated which often recognize nothing and receive a CCRR of 1.0. This causes many comparisons to be a tie which is not counted as a win for the correct antibody. An analysis of random antibodies scored based on dataset 1 of 500 confirmed this hypothesis with 99.74% having a CCRR of 1.0 when $P_{wc} = 0.1$.

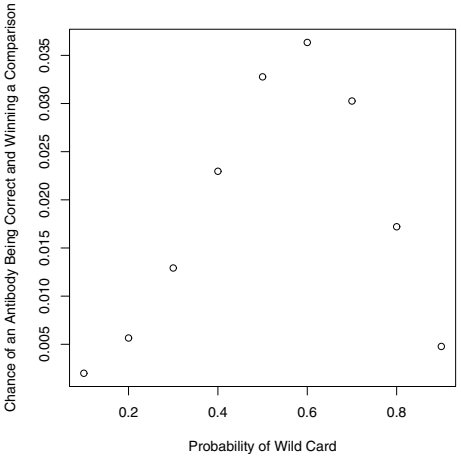


Fig. 4. Plot showing the chance that a random antibody will contain the correct model and win a comparison against a random antibody which does not contain the correct mode

4 Discussion and Conclusions

The goal of this paper was to show that the AIS was doing valuable work. First we compared it to a random search. The random search behaved exactly as expected; it relied on many antibodies containing rules for only $X0$ and $X1$ and few other attributes being randomly generated. This is very unlikely for a low P_{wc} setting and vice versa. On the other hand, we hypothesized that if our AIS was doing valuable work it would not need to rely on good antibodies randomly occurring. This was confirmed as our AIS performed the same as a random search on the highest P_{wc} setting, and performed better on each lower setting. This shows that the AIS was detecting the correct model within a larger antibody, and then paring down the antibody via mutation to a smaller and more accurate rule set. This is very promising as it shows that the AIS is capable of finding the correct model quickly. The random search is still NP time, as it requires a good antibody to show up by chance. Work done in Penrod et al. proved largely unsatisfactory compared to a random search [9]. This may have been caused by the misguided assumption that a high wild card probability was necessary for functioning. It may be interesting to note that one of the changes that caused the most immediate improvement in performance was to change the 'recognition' requirement. Namely to satisfaction for all the rules instead of only half; however, the authors are unsure as to exactly what significance this may have.

Next we showed that the AIS functioned better than expected by the ROC analysis. The ROC analysis correctly predicts that the power will increase as

we lower P_{wc} from 0.9 to 0.6. However, the AIS outperforms the ROC analysis' prediction for wild card probabilities lower than 0.6. This shows that the AIS does an excellent job of mutating a large antibody down to the correct model. Again this is promising, as it shows that the AIS is doing work in excess of that which a simplistic analysis predicts. In other words, the AIS is exhibiting complexity above both a random search and the more complicated ROC analysis.

We have shown that an AIS can at least compete with MDR on hard datasets, and do so in a similar computational time. We would like to point out that MDR does quite poorly on these datasets, failing more often than succeeding, despite being an exhaustive search. An analysis of when MDR and AIS fail, and whether the two are different, might be an interesting future project. We believe there are many ways one might potentially improve AIS to match or exceed the performance of MDR on difficult datasets, one of the simplest might be a reappraisal of the way we determine the solution attributes based on AIS output. One interesting test would be to perform the same ROC curve analysis but for several different scoring metrics and analyze the sensitivity-specificity graphs. In this way one might determine exactly what scoring metric is appropriate for different types of signal detection problems. There are also many algorithmic changes which could be made, including neighborhood testing, adding noise during testing, and developing different mutation strategies that have already been shown to increase the performance of AIS's.

In conclusion we have redesigned an AIS based on previous work by Penrod et al. This new AIS performs very well on difficult datasets. We have shown that an AIS works primarily by taking large noisy antibodies and pruning them until only the correct epistatic model is left. Perhaps most importantly we have shown that an AIS can detect non-linear epistatic signals with a much better efficiency than a random search; and that an AIS is more computationally efficient than MDR and is potentially a better solution for detecting epistatic signals in genetic datasets.

Acknowledgments. The authors would like to thank Todd Mackenzie for literature suggestion and Todd Mackenzie and Andrew Stella for a review of the manuscript.

References

1. Bereta, M., Burczynski, T.: Comparing binary and real-valued coding in hybrid immune algorithm for feature selection and classification of ECG signals. Eng. Appl. of AI, 571–585 (2007)
2. De Castro, L.N., Timmis, J.: Artificial immune systems: a new computational intelligence approach. Springer, Heidelberg (2002)
3. Emily, M., Mailund, T., Hein, J., Schauser, L., Schierup, M.H.: Using biological networks to search for interacting loci in genome-wide association studies. European Journal of Human Genetics 17, 1231–1240 (2009)
4. Fawcett, T.: ROC graphs: Notes and practical considerations for researchers. HP Labs Tech Report, HP Labs (2004)

5. Greene, C.S., White, B.C., Moore, J.H.: Ant Colony Optimization for Genome-Wide Genetic Analysis. In: Dorigo, M., Birattari, M., Blum, C., Clerc, M., Stützle, T., Winfield, A.F.T. (eds.) ANTS 2008. LNCS, vol. 5217, pp. 37–47. Springer, Heidelberg (2008)
6. Moore, J.H., Williams, S.M.: Epistasis and its Implications for Personal Genetics. The American Journal of Human Genetics 85, 309–320 (2009)
7. Moore, J.H.: Genome-wide analysis of epistasis using multifactor dimensionality reduction: feature selection and construction in the domain of human genetics. In: Knowledge Discovery and Data Mining: Challenges and Realities with Real World Data, pp. 17–30 (2007)
8. Motsinger, A.A., Lee, S.L., Mellick, G., Ritchie, M.D.: GPNN: power studies and applications of a neural network method for detecting gene-gene interactions in studies of human disease. BMC Bioinformatics 7, 39 (2006)
9. Penrod, N., Greene, C., Granizo-Mackenzie, D., Moore, J.H.: Artificial Immune Systems for Epistasis Analysis in Human Genetics. In: Pizzuti, C., Ritchie, M.D., Giacobini, M. (eds.) EvoBIO 2010. LNCS, vol. 6023, pp. 194–204. Springer, Heidelberg (2010)
10. Wilcoxon, F.: Individual Comparisons by Ranking Methods. Biometrics Bulletin 1, 80–83 (1945)
11. Velez, D.R., White, B.C., Motsinger, A.A., Bush, W.S., Ritchie, M.D., Williams, S.M., Moore, J.H.: A balanced accuracy function for epistasis modeling in imbalanced datasets using multifactor dimensionality reduction. Genetic Epidemiology 31, 306–315 (2007), doi:10.1002/gepi.20211

A Biologically Informed Method for Detecting Associations with Rare Variants

Carrie C. Buchanan[1,2], John R. Wallace[2], Alex T. Frase[2], Eric S. Torstenson[1], Sarah A. Pendergrass[2], and Marylyn D. Ritchie[2]

[1] Center for Human Genetics Research, Vanderbilt University, Nashville, TN
carrie.c.buchanan@vanderbilt.edu,
torstenson@chgr.mc.vanderbilt.edu
[2] Center for Systems Genomics, Pennsylvania State University, University Park, PA
{jrw32,atf3,sap29,marylyn.ritchie}@psu.edu

Abstract. With the recent flood of genome sequence data, there has been increasing interest in rare variants and methods to detect their association to disease. Many of these methods are collapsing strategies which bin rare variants based on allele frequency and functional predictions; but at this point, most have been limited to candidate gene studies with a small number of candidate genes. We propose a novel method to collapse rare variants based on incorporating biological information from the public domain. This paper introduces the functionality of BioBin, a biologically informed method to collapse rare variants and detect associations with a particular phenotype. We tested BioBin using low coverage data from the 1000 Genomes Project and discovered appropriate binning characteristics based on what one might expect given the size of the gene. We also tested BioBin using the pilot targeted exome data from 1000 Genomes Project. We used biologically-informed binning and differences in minor allele frequencies as a means to distinguish between two ancestral populations. Although BioBin is still in developmental stages, it will be a useful tool in analyzing sequence data and uncovering novel associations with complex disease.

Keywords: Rare Variants, Prior Knowledge, Collapsing Tool, Pathway Analysis.

1 Introduction

1.1 Paucity of Analytical Tools to Manage Sequence Data

Technological advances have dramatically increased the availability of genome sequence data at diminishing costs. We are hindered in exploiting these laboratory advances because strategies for analyzing these data to utilize their maximal potential are scarce. In fact, this wealth of data has made distinguishing true scientific discoveries from the thousands of false discoveries even more challenging. The opportunities for sequence-level association studies are abundant, including candidate gene, linkage studies, whole-exome sequencing, and whole-genome sequencing. The growing disparity in rapidly advancing data collection vs. slowly developing data analysis methods mandates a more concerted research effort to develop the necessary analytical tools to successfully interpret the genotypic and biologic data.

M. Giacobini, L. Vanneschi, and W.S. Bush (Eds.): EvoBIO 2012, LNCS 7246, pp. 201–210, 2012.

In the era of genome-wide association studies (GWAS), there was a focus on common variants, often missing valuable information about epistatic (gene-gene, GxG) and gene-environment (GxE) interactions with the DNA, structural variants, and rare variants[1]. While researchers have been able to reproducibly attribute common causal variants to over 80 diseases and traits[2], the estimated odds ratios are predominantly less than 1.5 and do not explain a large fraction of the estimated heritability. In an effort to elucidate heritability and to take advantage of the new sequencing technology, many researchers are investigating, in particular, the effects of rare variants. It is believed that rare variants can act alone, in concert with other rare variants, or together with common variants. There is increasing evidence to support a role for rare variants to contribute to risk of common, complex disease. Recent studies have implicated rare variants with moderate effect sizes using phenotypes such as obesity, autism, schizophrenia, hypertriglyceridemia, hearing loss, complex I deficiency, and type-1 diabetes[3-6]. Because association signals for rare variants are harder to detect and because they may act in concert, methods can be used to group the rare variants and test for group association with disease status. Potentially, multiple rare variants can account for missing heritability in a given trait.

Rare variants, which are more prevalent in sequencing studies, have low r^2 values and cannot be detected using a tag-SNP approach[7]. To date, most sequence analysis tools use standard analytical methods to reduce the search space. One standard method is to use family data which allows the analyst to exploit transmission patterns to filter the data[8]. This strategy is effective but not applicable to data sets without family information. Another technique is to perform a candidate gene study and collapse rare variants into bins in order to combine association signals. There are many explanations that describe why collapsing methods are favorable over other strategies:

1. Application to case-control studies
2. Application to whole-genome sequence data
3. Enrich association signals by combining otherwise undetectable rare variants
4. Reduce the degrees of freedom in the statistical test

The first published collapsing method, the cohort allelic sums test (CAST), calculates the sums of allelic mutation frequencies in cases versus controls and applies a statistical test to determine if the difference is statistically significant[9]. The CAST method assumes that rare variants have the same magnitude and direction of effect. Because the method uses a chi-square statistic, it is less than ideal because it does not easily incorporate covariates, cannot be used for quantitative phenotypes, and does not measure the direction of association[10]. Li and Leal developed a similar method, the combined multivariate and collapsing method (CMC), which uses a multivariate statistical test and permits combined analysis of rare and common variants[7]. The CMC method has improved power over CAST, presumably because functional information (i.e. direction of effect) was incorporated and because the method can be implemented in a regression framework[11]. Another option for collapsing is to weight the variants before collapsing based on some identified characteristic. Madsen and Browning proposed a collapsing method that weights each variant using its allele frequency and then performing a rank sum test between cases and controls [12]. Price et al. proposed a method to optimize the grouping of rare variants using a

variable-threshold approach based on allele frequency[13]. Several other methods cleverly incorporate functional data to guide collapsing and use a regression framework for statistical association[14,15]. One of the most promising collapsing methods, VAAST, was published more recently by Yandell et al. The VAAST algorithm evaluates each variant by allele frequency and assigns the variant functional prediction. It was shown to have better predictive power than SIFT and able to reliably identify disease-causing candidate genes[16]. However, there are two justifications why these methods must be improved in order to be useful for sequence data. First, few of these tools have been proven to manage whole-genome sequence data. This approach limits novel discovery since a particular set of genes or pathways must be selected for analysis. Second, incorporating functional information for collapsing is risky. Predictive function algorithms such as SIFT or PolyPhen are often incorrect[15], and creating a bin based simply on allele frequency makes an implicit assumption between allele frequency and odds ratio[13]. However, creating a weighted collapsing method incorporating multiple pieces of information spreads the risk and increases the likelihood that the aggregate is biologically similar.

1.2 Biofilter, a Tool to Uncover GxG and GxE Interactions

Biofilter was developed in our lab to reduce the search space in large scale GWAS studies[17]. Biofilter prioritizes interactions based on statistical and biological knowledge. It uses data from multiple sources that contain information about biological pathway and SNP interactions (see Table 1). Figure 1 shows a schematic of Biofilter.

Table 1. List of biological databases incorporated into Biofilter

Name	Databases
KEGG	Pathway maps for interactions, reactions, and relations
GO	Gene ontology and gene annotations
Reactome	Open reaction and pathway db, curated by expert biologists
DIP	Experimentally determined protein-protein interactions
NetPATH	Curated signal transduction pathways
PFAM	Protein family annotation and alignments
BioGrid	Curated interaction repository
MINT	Experimentally verified protein-protein interactions the literature
PharmGKB	Annotated gene variants and gene-drug-disease links
PharmSpresso*	Search engine for literature mining
BIND*	Stores interactions, molecular complexes and pathways
BioCARTA*	Metabolism pathways
HPRD*	Curated information on OMIM implicated proteins
UniPathway	Metabolic pathways for UniProtKB
MetaCyc*	Non-redundant metabolic pathways
NHGRI-GWAS	Catalog of SNP-trait associations
cisRED	Regulatory element predictions
ORegAnno*	Open regulatory annotation database
PolyPhen2	Predicts functional effects
TRRD*	Transcription regulatory regions database
ECRbase*	Evolutionary Conserved Regions database
UCSC*	UCSC Genome Bioinformatics

* denotes databases currently being integrated into Biofilter

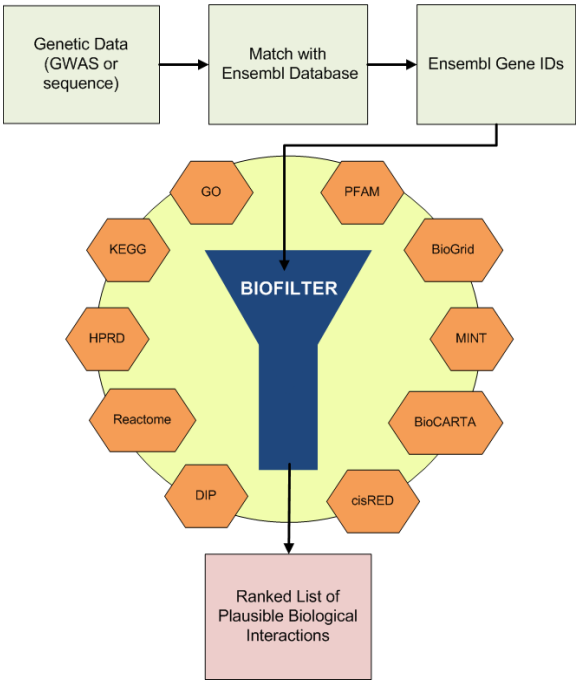


Fig. 1. Schematic of Biofilter

The Biofilter was developed to provide a mechanism to filter GWAS data based on known biology and allow for a comprehensive search for GxG interactions. However, the wealth of biological data in the database can be very informative for sequence data collapsing algorithms. Thus, it will serve as the database of biological knowledge for BioBin.

2 Methodology

2.1 Integrate Collapsing Method for Rare Variants Using Biological Knowledge

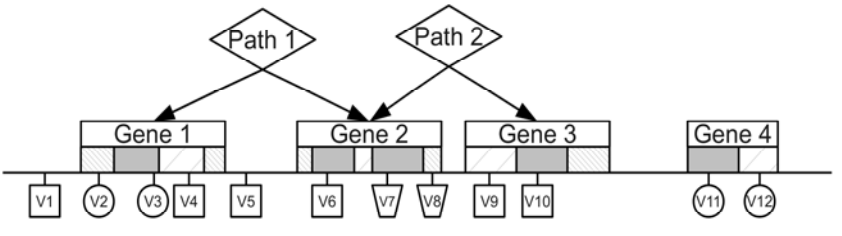
Existing collapsing strategies bin primarily on allele frequency and functional prediction; however, these have mainly been limited to candidate gene studies. Incorporating biological information into a collapsing strategy is useful because it can be expanded to whole-genome data, serves a purpose for data reduction in these large data sets, and can be used as a framework to include expression data in the future. Our first goal was to successfully develop a flexible model using biological information. Currently, BioBin incorporates two pieces of information to bin rare variants:

1. Allele frequency threshold (set as parameter by user)
2. Biological domain knowledge (i.e. gene boundaries, pathway groups)

In BioBin the allele frequencies are calculated from the data and used to categorize rare and common variants based on the user's frequency threshold. This is not identical to the variable allele frequency threshold described by Price et al.; however, it does address similar concerns since the threshold is a flexible parameter[13].

The most important contribution of BioBin is the incorporation of biological knowledge. Users of BioBin are able to specify biological boundaries for creating bins. For example, the user can specify that bins are bounded by pathways, gene boundaries, or sub-gene elements (introns, exons, etc) without the necessity of an external feature file. Alternatively, the user can choose to reduce the search space by limiting analysis to specific databases of information, biological pathways, or implicated genes. BioBin collapses variants based on prior domain knowledge in a way that is meaningful for biological interpretation. It is possible to over-fit the data using prior knowledge. To address this, one should incorporate permutation testing on disease status among affected and unaffected individuals.

Users have the flexibility of determining the feature level for each analysis (see detailed structure in Figure 2). The user chooses the burden test to determine the bin boundaries (pathway, gene, sub-gene, etc). As the limiting factor of subdividing bins is an issue of statistical power, there will be variant thresholds that will internally maintain the correct number of variants in a bin.



Legend	
Gene Information	
	Exon
	Intron
	Regulatory
Variant Information	
	Protective
	Detrimental
	Neutral (Unk)

Sample Binning Strategies

Gene Burden Analysis

Gene 1	Gene 2	Gene 3	Gene 4	Intergenic
V2, V3, V4	V6, V7, V8	V9, V10	V11, V12	V1, V5

Pathway Burden Analysis

Path 1	Path 2	Gene 4	Intergenic
V2, V3, V4, V6, V7, V8	V6, V7, V8, V9, V10	V11, V12	V1, V5

Functional Pathway Burden Analysis

Path 1 (+)	Path 1 (-)	Path 1 (N)	Path 2 (-)	Path 2 (N)	Gene 4 (+)	Intergenic (N)
V2, V3	V7, V8	V4, V6	V7, V8	V9, V10	V11, V12	V1, V5

Fig. 2. Sample binning structure and strategies. Pathways are referred to as "Path X" and "V" refers to genetic variants. +, -, N corresponds to functional binning based on protective, detrimental, and neutral variants.

Figure 2 shows an example containing a small number of genes and pathways and a sampling of the many ways that variants can be binned together. Unlike other methods, BioBin captures intergenic variants, potentially discovering novel

associations outside of currently known gene boundaries. Additionally, as the example shows, the number of bins grows rapidly with increasing amounts of knowledge used, which demonstrates the importance of BioBin's ability to prioritize the bins. BioBin allows a user to assign weights to a particular interesting group of bins and ranks the bins according to the user preference, so the user need only test the top percentage of bins for association, thereby reducing the penalty for multiple testing and increasing the chances of finding a true significant association.

In recent publications concerning collapsing rare variants, the complexity of models tested has been variable. There is a penalty associated with models that include unanticipated parameters; for example, the CMC method is less powerful when variants of opposite direction of effect are included because the study design did not account for opposing effects[11]. However, the functional significance of a given variant is incredibly important if one is trying to discover an association between a regional bin of variants and a phenotype. The most effective method would be to include only variants that demonstrate the same functional effect. This has improved power in simulated studies and will be implemented in BioBin using a similar prediction algorithm. This option will increase memory required, but will also increase the chance to find an association with bins that are functionally similar.

2.2 Initial Testing

Distribution of Variants Across Low Coverage Data from 1000 Genomes Project. Method testing can be problematic, particularly for sequence data driven tools. For BioBin it is important to consider the validity of the code, its capacity to handle large data sets, and applicability to detecting genetic associations. As a way to test validity and data capacity, we used CEU (population originating from Northwestern Europe) low-coverage pilot data from the 1000 Genomes Project[18]. This data set contains whole-genome sequence data from 60 unrelated CEU individuals with an average depth of coverage between 2x-6x.

For this analysis, the most terminal nodes (bins) were created using physical gene boundaries. We hypothesized that the number of variants in any given bin should correlate with the size of the gene. Therefore, an approximately increasing linear relationship between the number of variants and gene size should be apparent in the data.

Case-Control Analysis Using Targeted Exome Data from 1000 Genomes Project. Second, we tested the ability of BioBin to produce results compatible with an association study. Using 1000 Genomes targeted exome pilot data, we borrowed an idea from population genetics. A similar method was published by Madsen and Browning in 2009. To mimic disease resequencing, they grouped exonic rare variants by each region and compared across YRI (Yoruba people of Ibadan, Nigeria) and CEU (people of northern and western European ancestry) populations. The authors used five 100 kb regions of sequenced Encode III data in two populations and performed a case-control analysis using population identity as the phenotype[12]. This is a favorable approach since it utilizes natural data without the need for simulation and takes advantage of natural differences in allele frequencies between two populations. For this study, we used 1000 Genomes exome pilot data composed of 90 CEU

samples and 66 TSI samples (Tuscan Italians from Southern Europe). The 1000 Genomes exome pilot targeted 8,140 exons from 906 randomly selected genes. It is high quality data with a depth of coverage >70x for both CEU and TSI[18]. We performed a Fisher's Exact Test on the BioBin results to determine association with population identity.

3 Application

For the initial test of BioBin, we used 1000 Genomes Project pilot data from the CEU population. One would reasonably expect a strong correlation between the rare variant distribution and the size of the gene. Using an allele frequency threshold of 0.02 (which specifies the rare/common variant threshold), BioBin evaluated the low coverage data. On a desktop machine with 12 GB RAM and Intel Xeon Processor (2.2 GHz), the run time for processing this data was approximately two hours and used 7.4 GB of memory. On a high performance cluster, using a single node with 3.07GHz processor, the run time was approximately 45 minutes. Figure 3 shows the relationship between the bin sizes versus the gene size. The corresponding fitted line is shown on Figure 3 in red (r=0.6750, p-value < 0.0001).

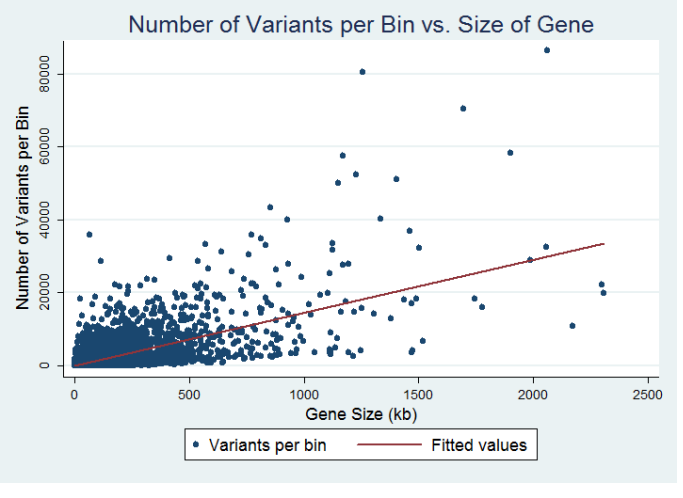


Fig. 3. Number of variants in each bin plotted against gene size across the entire genome

Using the smaller exome data set, we tested the collapsing algorithm using "phenotype" data. The pilot high coverage exome data from 1000 Genomes Project includes targeted exome data for 156 CEU and TSI individuals. We dichotomized the independent variable by the presence or absence of rare variants and used population identity to code case-control status (dependent variable). The statistical analysis was completed using a Fisher's exact test. However, this analysis is amenable to a regression framework by utilizing the proportion of rare variants present instead of the all-or-nothing approach.

The bins accounted for 762 genes. P-values were appropriately compared to a corrected threshold of significance using Bonferroni's conservative correction for multiple testing ($0.05/762=6.6X10^{-5}$). Table 2 shows the top genes that differ between the two populations, p-values, genomic location, and their annotated function.

Table 2. Genes associated with population identity

Gene	P-value*	Function
CDHR5	$<10^{-7}$**	Cadherin-related family member 5
HNRNPUL1	0.007	Heterogeneous nuclear ribonucleoprotein U-like 1
PLAT	0.03	Plasminogen activator, tissue

*P-values calculated from Fisher's Exact Test
** Significant after correction using Bonferroni correction for multiple testing

For more clarification, Table 3 shows the contingency table used in the analysis of CDHR5, the only remaining significant gene after multiple testing correction.

Table 3. Contingency table for CDHR5

	CDHR5		
Population	**No variants**	**At least one variant**	**Total**
CEU	38	52	90
TSI	3	63	66
Total	41	115	156

4 Discussion

BioBin is in early developmental stages, but exhibits quite a bit of promise as a flexible collapsing tool. It can be implemented into an analysis plan for studying complex phenotypes. This tool serves the purpose of data reduction, simplifying pathway analyses, and offers a plausible framework for integrating large scale complex data sets into a single analysis.

The preliminary results using 1000 Genomes Pilot data are satisfactory. The initial challenges were to successfully read in and manipulate variant call format (VCF) files, computationally build and store "trees" of biological information from which we could form bins, and determine the most useful output that could be used in a statistical analysis pipeline.

As a test of validity, we plotted the number of variants in each bin with the size of the gene. Figure 3 shows a statistically significant linear trend in the data. As expected, increasing bin sizes were correlated with increasing gene sizes.

BioBin will ultimately be used for association discovery in case-control analyses. One can use bins as a means of data reduction for sequence data, then test the aggregate bins for phenotype association. We performed a case-control analysis using 1000 Genomes Project exome pilot data, where ethnicity was considered a binary dependent variable. There was one gene in the pilot data that was statistically

significant and thus could be used to distinguish between CEU and TSI populations using differences between binned variants. To our knowledge, variants in CDHR5 have not been previously associated with population identity.

5 Conclusion

BioBin is a novel collapsing method that uses allele frequency data and biological information to bin rare variants. The resultant bins can be tested in a regression framework for association with a given phenotype. The advantages of using a biologically-informed method are:

1. Capability for whole-genome and whole-exome analyses
2. Practical method for data reduction in sequence data analysis
3. Utilizes domain knowledge to prioritize results for association testing
4. Accurate binning increases the statistical power to detect associations
5. Output can be used in Biofilter to identify GxG, GxE, and gene-drug interactions
6. Can be combined with common variant methods
7. Provides framework to integrate numerous types of complex data sets

These results demonstrate that BioBin is capable of handling large data sets, the bin-variant distribution agrees with our expectation given gene sizes, and that rare-variant analyses can be done with case-control studies. While the development of BioBin is far from complete, it will be a useful tool for researchers studying complex disease.

References

1. Eichler, E.E., Flint, J., Gibson, G., Kong, A., Leal, S.M., Moore, J.H., Nadeau, J.H.: Missing heritability and strategies for finding the underlying causes of complex disease. Nat. Rev. Genet. 11(6), 446–450 (2010)
2. Hindorff, L.A., Sethupathy, P., Junkins, H.A., Ramos, E.M., Mehta, J.P., Collins, F.S., Manolio, T.A.: Potential etiologic and functional implications of genome-wide association loci for human diseases and traits. Proc. Natl. Acad Sci. U S A 106(23), 9362–9367 (2009)
3. Johansen, C.T., Wang, J., Lanktree, M.B., Cao, H., McIntyre, A.D., Ban, M.R., Martins, R.A., Kennedy, B.A., Hassell, R.G., Visser, M.E., Schwartz, S.M., Voight, B.F., Elosua, R., Salomaa, V., O'Donnell, C.J., Dallinga-Thie, G.M., Anand, S.S., Yusuf, S., Huff, M.W., Kathiresan, S., Hegele, R.A.: Excess of rare variants in genes identified by genome-wide association study of hypertriglyceridemia. Nat. Genet. 42(8), 684–687 (2010)
4. Walsh, T., Shahin, H., Elkan-Miller, T., Lee, M.K., Thornton, A.M., Roeb, W., Abu, R.A., Loulus, S., Avraham, K.B., King, M.C., Kanaan, M.: Whole exome sequencing and homozygosity mapping identify mutation in the cell polarity protein GPSM2 as the cause of nonsyndromic hearing loss DFNB82. Am. J. Hum. Genet. 87(1), 90–94 (2010)
5. Bhatia, G., Bansal, V., Harismendy, O., Schork, N.J., Topol, E.J., Frazer, K., Bafna, V.: A covering method for detecting genetic associations between rare variants and common phenotypes. PLoS Comput. Biol. 6(10), e1000954 (2010)
6. Ionita-Laza, I., Buxbaum, J.D., Laird, N.M., Lange, C.: A new testing strategy to identify rare variants with either risk or protective effect on disease. PLoS Genet. 7(2), e1001289 (2011)

7. Li, B., Leal, S.M.: Methods for detecting associations with rare variants for common diseases: application to analysis of sequence data. Am. J. Hum. Genet. 83(3), 311–321 (2008)
8. Haack, T.B., Danhauser, K., Haberberger, B., Hoser, J., Strecker, V., Boehm, D., Uziel, G., Lamantea, E., Invernizzi, F., Poulton, J., Rolinski, B., Iuso, A., Biskup, S., Schmidt, T., Mewes, H.W., Wittig, I., Meitinger, T., Zeviani, M., Prokisch, H.: Exome sequencing identifies ACAD9 mutations as a cause of complex I deficiency. Nat. Genet. 42(12), 1131–1134 (2010)
9. Morgenthaler, S., Thilly, W.G.: A strategy to discover genes that carry multi-allelic or mono-allelic risk for common diseases: a cohort allelic sums test (CAST). Mutat. Res. 615(1-2), 28–56 (2007)
10. Bansal, V., Libiger, O., Torkamani, A., Schork, N.J.: Statistical analysis strategies for association studies involving rare variants. Nat. Rev. Genet. 11(11), 773–785 (2010)
11. Basu, S., Pan, W.: Comparison of Statistical Tests for Association with Rare Variants. 1-33. 11-30-2010. Research report, Division of Biostatistics, University of Minnesota
12. Madsen, B.E., Browning, S.R.: A groupwise association test for rare mutations using a weighted sum statistic. PLoS Genet. 5(2), e1000384 (2009)
13. Price, A.L., Kryukov, G.V., de Bakker, P.I., Purcell, S.M., Staples, J., Wei, L.J., Sunyaev, S.R.: Pooled association tests for rare variants in exon-resequencing studies. Am. J. Hum. Genet. 86(6), 832–838 (2010)
14. Han, F., Pan, W.: A data-adaptive sum test for disease association with multiple common or rare variants. Hum. Hered. 70(1), 42–54 (2010)
15. Hoffmann, T.J., Marini, N.J., Witte, J.S.: Comprehensive approach to analyzing rare genetic variants. PLoS One 5(11), e13584 (2010)
16. Yandell, M., Huff, C., Hu, H., Singleton, M., Moore, B., Xing, J., Jorde, L.B., Reese, M.G.: A probabilistic disease-gene finder for personal genomes. Genome Res. 21(9), 1529–1542 (2011)
17. Bush, W.S., Dudek, S.M., Ritchie, M.D.: Biofilter: a knowledge-integration system for the multilocus analysis of genome-wide association studies. In: Pac. Symp. Biocomput., pp. 368–379 (2009)
18. Durbin, R.M., Abecasis, G.R., Altshuler, D.L., Auton, A., Brooks, L.D., Durbin, R.M., Gibbs, R.A., Hurles, M.E., McVean, G.A.: A map of human genome variation from population-scale sequencing. Nature 467(7319), 1061–1073 (2010)

Complex Detection in Protein-Protein Interaction Networks: A Compact Overview for Researchers and Practitioners

Clara Pizzuti[1], Simona E. Rombo[1,2], and Elena Marchiori[3]

[1] Institute for High Performance Computing and Networking,
National Research Council of Italy, CNR-ICAR,
via P. Bucci 41C, 87036 Rende (CS), Italy
pizzuti@icar.cnr.it
[2] DEIS, Università della Calabria
via P. Bucci 41C, 87036 Rende (CS), Italy
simona.rombo@deis.unical.it
[3] Radboud University, Department of Computer Science
Nijmegen, The Netherlands
elenam@cs.ru.nl

Abstract. The availability of large volumes of protein-protein interaction data has allowed the study of biological networks to unveil the complex structure and organization in the cell. It has been recognized by biologists that proteins interacting with each other often participate in the same biological processes, and that protein modules may be often associated with specific biological functions. Thus the detection of protein complexes is an important research problem in systems biology. In this review, recent graph-based approaches to clustering protein interaction networks are described and classified with respect to common peculiarities. The goal is that of providing a useful guide and reference for both computer scientists and biologists.

1 Introduction

In the last few years the development of advanced high-throughput technologies [48] to determine protein interactions has made available large volumes of experimental data that reflect the interplay among proteins in complex cellular networks. *Protein-protein interaction (PPI) networks* can be used for discovering (putative) functional modules, or complexes, consisting of proteins sharing a common function. This is motivated by the observation that proteins are organized into different putative protein complexes each performing specific tasks in the cell [18,36] and that proteins interacting with each other often participate in the same biological processes. Furthermore, protein modules can often be associated with specific biological functions and proteins belonging to a specific module are more related to each other than to the members of other modules [47]. Therefore the detection of putative protein complexes using PPI networks can help in understanding the mechanisms regulating cell life, in describing the

M. Giacobini, L. Vanneschi, and W.S. Bush (Eds.): EvoBIO 2012, LNCS 7246, pp. 211–223, 2012.

evolutionary orthology signal (e.g., [22]), in predicting the biological functions of uncharacterized proteins, and, more importantly, for therapeutic purposes.

It is worth pointing out that protein complexes and functional modules have different biological meanings. A protein complex is a molecular machine that consists of several proteins that bind each other at the same place and time. On the contrary, a functional module consists of a few proteins that control or perform a particular cellular function through interactions between themselves (these proteins do not necessarily interact at the same time and place). However, it is hard to distinguish them in many cases because analyzed pair-wise protein interactions do not have temporal and spatial information, thus in the following we will use the two terms as synonyms.

The problem of detecting protein complexes using PPI networks can be computationally addressed by using clustering techniques. Clustering consists of grouping data objects into groups (clusters) such that the objects in the same cluster are more similar each other than with objects in the other clusters [20]. In PPI networks, clustering means grouping together proteins which share a large number of interactions. These clusters are considered to represent functional modules. Possible uncharacterized proteins in a cluster may be assigned to the biological function recognized for that module. PPI networks have various characteristics which have to be taken into account when developing clustering algorithms for detecting functional complexes. Therefore, a number of clustering approaches have been proposed to extract relevant modules from PPI networks.

In this work, we present a short overview of state-of-the-art clustering methods for complex detection in PPI networks, by introducing a classification criterion that is different from those proposed previously. We mainly focus on methods that use only the topology of the graph for detecting clusters, and do not employ similarity measures between proteins as described by vectors of features (for instance, features derived by the protein aminoacid sequences or by functional domain composition of proteins). Our goal is twofold: (a) to guide researchers in the development of new methods for clustering PPI networks by providing a description of the main algorithmic approaches of state-of-the-art methods; and (b) to guide practitioners in the application of methods by providing information about their availability.

In this respect our contribution differs from that contained in other surveys, whose main goal is either to describe and compare experimentally methods presented in the literature, such as [2,8,40,28,41,49,27], or to highlight the computational aspects of graph-based analysis of networks [34].

2 Methods

Clustering approaches for detecting protein complexes in PPI networks can be broadly categorized as distance-based and graph-based ones [28]. Distance-based clustering approaches employ the concept of distance between two proteins as described by vectors of features (for instance, derived by their aminoacid sequence) [7,43,4,35]. Graph-based clustering techniques (mainly) consider the topology of

the network. These latter techniques are deeply studied in other research fields, such as physics and data mining, and are known as community detection methods [17].

We distinguish the following five main types of algorithmic approaches employed in methods for complex detection in PPI networks:

1. Local neighbourhood Density search (LD);
2. Cost-based Local search (CL);
3. Flow Simulation (FS);
4. Statistical-based Measures (SM);
5. Population-based Stochastic search (PS).

For each of the categories listed above, we describe a selection of methods by focusing on those that can be directly used by practitioners, that is, whose software is publicly available.

2.1 Local Neighborhood Density Search (LD)

Many methods, including the most popular, are based on local neighbourhood density search. Their objective is to find dense subgraphs (that is, each node is connected to many other nodes in the same subgraph) within the input network. We summarize in the following six representative methods of this approach, and include a pointer to the software when publicly available.

One of the most popular methods for finding modules in PPI networks based on the LD approach is **MCODE** [6]. This method employs a node weighting procedure by local neighbourhood density and outward traversal from a locally dense seed protein, in order to isolate the dense regions according to given input parameters. The algorithm allows fine-tuning of clusters of interest without considering the rest of the network and allows examination of cluster interconnectivity, which is relevant for protein networks. It is implemented as Cytoscape plug-in. With a user-friendly interface, it is suited for both computationally and biologically oriented researchers.
`http://baderlab.org/Software/MCODE`.

In [3] the **DPClus** method for discovering protein complexes in large interaction graphs was introduced. It is based on the concepts of *node weight* and *cluster property* which are used for selecting a seed node to be expanded by iteratively adding neighbours, and to terminate the expansion process, respectively. Once a cluster is generated, its nodes are removed from the graph and the next cluster is generated using only the remaining nodes until all the nodes have been assigned to a cluster. The algorithm allows also to generate overlapping clusters by keeping the nodes already assigned to clusters.
`http://kanaya.naist.jp/DPClus/`.

SWEMODE was introduced in [30]. It identifies dense sub-graphs by introducing two network measures that combine functional information with topological properties of the networks. These measures, weighted cluster coefficient

and weighted nearest-neighbours degree, compute the strengths of interactions between the proteins by using their semantic similarity based on the Gene Ontology terms of the proteins.
`No publicly available implementation.`

DECAFF [26], is an algorithm to mine protein complexes in PPI networks that tries to address two major limitations plaguing protein interaction data, namely incompleteness and noise. The method consists of three main steps: detection of local dense neighbourhoods of each protein, merging of the local sub-graphs on the base of the similarity degree between neighbourhoods, filtering away possible false complexes detected.
`No publicly available implementation.`

CFinder is a program for detecting and analyzing overlapping dense groups of nodes in networks; it is based on the clique percolation concept (see [12,33,1]). The idea behind this method is that a cluster can be interpreted as the union of small fully connected sub-graphs that share nodes, where a parameter is used to specify the minimum number of shared nodes.
`http://hal.elte.hu/cfinder/wiki/?n=Main.Manual`.

The greedy local expansion method **PINCoC** was introduced in [38]. It expands a single protein randomly selected by adding/removing connected proteins that best contribute to improve a given quality function based on the concept of co-clustering [32] that favors the detection of maximal dense groups. In order to escape poor local maxima, with a given probability, the protein causing the minimal decrease of the quality function is removed. An extension of PINCoC for detecting multi-functional protein complexes, called MF-PINCoC, was introduced in [39].
`http://wwwinfo.deis.unical.it/~rombo/pincoc/download.html`.

PCP is a method proposed in [11] that exploits the shared interaction partners of proteins, i.e., the level-2 neighbours. The method transforms the input graph by adding edges between level-2 neighbours and by removing edges, using a criterion that quantifies the likelihood that the two proteins of an edge share functions. Any clustering method can then be applied to the resulting graph. The authors proposed a clustering method that iteratively merges dense sub-graphs.
`http://www.comp.nus.edu.sg/~wongls/projects/complexprediction/PCP-3aug07/`.

DME [16] is a method for extracting dense modules from a weighted interaction network. The method detects all the node subsets that satisfy a user-defined minimum density threshold. The method returns only locally maximal solutions, i.e. modules where all the direct supermodules (containing one additional node) do not satisfy the minimum density threshold. The obtained modules are ranked according to the probability that a random selection of the same number of nodes produces a module with at least the same density. An interesting property of this method is that it allows to incorporate constraints with respect to additional data sources.

http://people.kyb.tuebingen.mpg.de/georgii/dme.html.

The methods based on the LD approach here briefly described have as common objective that of finding dense subgraphs within the network and to maximize the density of each subgraph.

MCODE and DPClus adopt a rather similar search strategy. They define the weight of each node, the node with highest weight is chosen as seed cluster, and neighbouring nodes are added to the current cluster if threshold parameters are satisfied. The main difference between the methods lies in the definition of weight.

The originality of PCP mainly relies in the procedure for transforming an interaction graph by adding and removing edges.

Both CFinder and the extended version of PINCoC, generate overlapping clusters, and use the concepts of k-clique and co-cluster to find dense subgraphs, respectively.

DME is somewhat different from all other methods since it enumerates *all* node subsets that satisfy a user-defined minimum density threshold. Each of the above mentioned methods require setting the values of some parameters; this influences the number and resolution of the discovered clusters. Other recent algorithms based on this approach include SPICi [23] and DEEN [21], two seed-based fast algorithms for complex detection in PPI networks.

2.2 Cost-Based Local Search (CL)

Methods based on cost-based local search extract modules from the interaction graph by partitioning the graph into connected subgraphs, using a cost function for guiding the search towards a best partition. We describe here in short three methods based on this approach with different characteristics.

A typical instance of this approach is **RNSC** [24], which explores the solution space of all the possible clusterings in order to minimize a cost function that reflects the number of inter-cluster and intra-cluster edges. The algorithm begins with a random clustering, and attempts to find a clustering with best cost by repeatedly moving one node from a cluster to another one. A list of tabular moves is used to forbid cycling back to previously examined solutions. In order to output clusters likely to correspond to true protein complexes, thresholds for minimum cluster size, minimum density, and functional homogeneity must be set. Only clusters satisfying these criteria are given as the final result. This obviously implies that many proteins are not assigned to any cluster.
http://www.cs.toronto.edu/~juris/data/rnsc/.

Several community discovery algorithms have been proposed based on the optimization of a modularity-based function (see e.g. [15]). Modularity measures the fraction of edges falling within communities, subtracted by what would be expected if the edges were randomly placed. In particular, **Qcut** [44] is an efficient heuristic algorithm applied to detect protein complexes. Qcut optimizes modularity by combining spectral graph partitioning and local search. By op-

timizing modularity, communities that are smaller than a certain scale or have relatively high inter-community density may be merged into a single cluster. In order to overcome this drawback, the authors introduce an algorithm that recursively applies Qcut to divide a community into sub-communities. In order to avoid over-partitioning, a statistical test is applied to determine whether a community indeed contains intrinsic sub-community.
http://cs.utsa.edu/~jruan/Software.html

Recently, the notion of **ModuLand** [25], has been introduced. **ModuLand** is an integrative method family for determining overlapping network modules as hills of an influence function-based, centrality-type community landscape, and including several widely used modularization methods as special cases. Several algorithms obtained from ModuLand provide an efficient analysis of weighted and directed networks, determine overlapping modules with high resolution, uncover a detailed hierarchical network structure allowing an efficient, zoom-in analysis of large networks, and allow the determination of key network nodes. It is implemented as Cytoscape plug-in.
http://www.linkgroup.hu/modules.php

2.3 Flow Simulation (FS)

Methods based on the flow simulation approach mimic the spread of information on a network. We report four methods based on this approach. The first two are based on the concept of random walk and are popular methods with available software. The other two methods exploit biological knowledge for passing information between proteins in the network in order to cluster proteins. Unfortunately, we could not find publicly available software for these two methods.

One of the first flow simulation method for detecting protein complexes in a PPI network is the *Markov Clustering algorithm MCL* [13]. **MCL** simulates the behaviour of many walkers starting from the same point, that move within the graph in a random way.
http://micans.org/mcl/

A more recent method based on flow simulation is **RRW** [31]. RRW is an efficient and biologically sensitive algorithm based on repeated random walks for discovering functional modules, which implicitly makes use of network topology, edge weights, and long range interactions between proteins.
http://www.cs.ucsb.edu/~kpm/software.html

IFB [10] proposed an Information Flow-Based approach to identify overlapping functional modules. The algorithm integrates topological and biological knowledge to select a number of informative proteins and simulates the information flow through the network from each informative protein. The weighted degree of a node is defined as the sum of the weights of the edges containing that node, and the weight of an edge is computed using the correlation between the

expression profiles of the two genes encoding the proteins linked by that edge. This weighted degree provides the semantic information of a node.
`No publicly available implementation.`

An interesting method based on flow simulation is **STM** [19], which finds clusters of arbitrary shape by modelling the dynamic relationships between proteins of a PPI network as a signal transduction system. The overall signal transduction behaviour between two proteins of the network is defined in order to evaluate the perturbation of one protein on the other one, both biologically and topologically. The signal transduction behaviour is modelled using the Erlag distribution.
`No publicly available implementation.`

2.4 Statistical Measures (SM)

The two following approaches rely on the use of statistical concepts to cluster proteins. They are based on the number of shared neighbours between two proteins, and on the notion of preferential attachment of the members of a module to other elements of the same module, respectively.

Samantha and Liang [45] proposed a clustering method, here called **SL** by the names of the authors, based on the idea that if two proteins share a number of common interaction partners larger than what would be expected in a random network, then they should be clustered together. The method assesses the statistical significance of forming shared partnership between a pair of proteins using the concept of p-value of a pair of proteins.

The p-values of all proteins pairs are computed and stored in a similarity matrix. The protein pair with the lowest p-value is chosen to form the first group and the corresponding rows and columns of the matrix are merged in a new row and column. The new p-value of the merged row/column is the geometric mean of the separate p-values of the corresponding elements. This process is repeated by adding new proteins to the actual cluster until a threshold is reached. The process is repeated on the remaining proteins until all the proteins have been clustered.
`No publicly available implementation.`

In [14] a statistical approach for the identification of protein clusters is presented, here called **Farutin** (the name of the first author). This method is based on the concept of preferential interaction among the members of a module. The authors use a novel metric to measure the community strength. The community strength is gauged by the preferential attachment of each member of a module to the other elements of the same module. This concept of preferential attachment is quantified by how unlikely it is observed in a random graph.

Since it is necessary to count the number of edges in the graph, the authors assume a random graph as the null model where an edge is the random variable. This measure of community strength is local, since it is a function of the subgraph induced by a set of proteins and their degrees. To identify the clusters a greedy approach that searches for a set of nodes in the network with small values

of community strength is adopted. At the beginning a list of two adjacent nodes is considered. The list is then grown by adding the node that leads to the largest decrease of the community score until no such node exists. This is repeated for each connected node pair, thus the obtained clusters can partially overlap.
`No publicly available implementation.`

2.5 Population-Based Stochastic Search (PS)

Population-based stochastic search has been used for developing algorithms for community detection in networks (see, e.g., [46,37]). However, we are aware of only two works that apply this approach to detect protein complexes in PPI networks.

Specifically, in [29] the authors proposed an algorithm based on evolutionary computation, here called **CGA**, for enumerating maximal cliques and apply it to the Yeast genomic data. The advantage of this method is that it can find as many potential protein complexes as possible.
`No publicly available implementation.`

Recently, in [42] an immune genetic algorithm, here called **IGA**, is described to find dense subgraphs based on efficient vaccination method, variable-length antibody schema definition and new local and global mutations. The algorithm is applied to clustering protein-protein interaction networks.
`No publicly available implementation.`

3 Discussion

We summarize the characteristics of each method in Table 1, with respect to few features: the structure of the clusters found by a method, the kind of approach it uses, whether the clusters are found simultaneously or one at a time, the capability of the method to detect overlapping clusters, if the method assigns each protein to a cluster, and if software for that method is publicly available.

All the considered methods have some input parameters that influence the number of clusters produced, the size, the density, and the structure. The LN methods, except CFinder, obtain the modules one at a time because they select a seed node and expand it until a condition, generally related to cluster density, is satisfied. Thus they can be considered bottom-up approaches: individual nodes are grouped together until all the graph has been examined. Methods that simultaneously find the clusters can be considered top-down. They consider the whole graph and try to partition it in connected components. Because of the threshold constraints incorporated in many methods in order to decide when a group of connected nodes is a cluster, nodes with few interactions are often discarded.

The elimination of sparsely connected nodes could result in the elimination of important information on the network structure and possibly prevent the detection of clusters of different topological shapes. Nevertheless, it is not clear

Table 1. Summary of some characteristics of the methods. The first column report the method acronym and reference, in chronological order. The second column reports the topological structure a method searches (a = arbitrary, d = dense sub-graphs). The approach each method is based on is reported in the third one. The fourth column (Simult.) specifies if the method finds all clusters simultaneously and the fifth column (Overlap) reports if the method generates overlapping clusters. Finally, the last two columns specify if the method returns some unassigned proteins (Un. Prot), and if software implementing that method is (publicly) available (Software).

Method	Structure	Approach	Simult.	Overlap	Un. Prot.	Software
MCL [13]	a	FS	yes	no	no	yes
SL [45]	a	SM	no	no	no	no
MCODE [6]	d	LN	yes	no	yes	yes
RNSC [24]	d	CL	yes	no	yes	yes
STM [19]	a	FS	yes	yes	yes	no
SWECODE [30]	d	LN	no	no	yes	no
DPClus [3]	d	LN	yes	no	yes	yes
IFB [10]	a	FS	no	yes	yes	no
Farutin [14]	a	SM	no	yes	no	no
CFinder [1]	d	LN	yes	yes	yes	yes
CGA [29]	d	PS	yes	yes	yes	no
PCP [11]	d	LN	no	yes	yes	yes
DECAFF [26]	d	LN	no	yes	yes	no
MF-PINCoC [38]	a	LN	no	yes	no	yes
Qcut [44]	d	CL	yes	no	no	yes
DME [16]	d	LN	no	yes	yes	yes
RRW [31]	a	FS	yes	no	no	yes
ModuLand [25]	d	CL	yes	yes	no	yes
IGA [42]	d	PS	yes	yes	no	no

whether the assumption that each protein has to belong to a cluster (representing a putative protein complex) is realistic, given the actual incompleteness of the PPI network data available, and forcing every node into a community could distort results [51].

Several challenges for the topic discussed in this work are still open. Notably among them, the necessity of diminishing the clustering methods dependence on many input parameters. Further improvements could be achieved by making a method able to set automatically some of its parameters, for example according to the density and/or characterization of the input PPI network.

Another interesting issue is that of finding a suitable compromise between the accuracy of the proposed method, and the portion of input graph that is involved in the final clustering. Indeed, the most accurate clustering methods are often able to assemble only a small percentage of the PPI network they analyze (e.g., **MCODE** [6]).

Furthermore, biological graphs are affected by inaccuracy, also due to the methods exploited in order to discover protein-protein interactions (e.g., high

throughput and computational methods). Although several techniques are able to exploit the specific reliability indices provided by the available interaction datasets (e.g., MINT [9]) as suitable filters during the clustering process, many efforts are still needed to make the clustering techniques more robust to such kind of noise.

Finally, all the considered methods, with the exception of **SWEMODE** [30], cluster the input biological graph only on the basis of topological connections. An interesting challenge would be that of combining the main advantages of the considered approaches with taking into account also possible properties of the nodes, such as protein sequence similarity, Gene Ontology annotations [5] or functional domain composition of proteins [50].

4 Conclusion

In this paper, we presented a compact survey of graph-based clustering methods for detecting protein complexes in PPI networks. We proposed a classification based on five main categories, that are, local neighbourhood density search, cost-based local search, flow simulation, statistical measures and population-based stochastic search. We summarized the main algorithmic features and software availability of the considered methods, by also discussing their possible limitations. Finally, we pointed out some open issues related to the problem of clustering PPI networks.

We hope that the overview presented in this paper will be used by both computer scientists and practitioners as a quick reference for guiding the selection, use and development of algorithms for discoverying protein complexes and functions through the analysis of PPI networks.

References

1. Adamcsek, B., Palla, G., Farkas, I.J., Dernyi, I., Vicsek, T.: Cfinder: locating cliques and overlapping modules in biological networks. Bioinformatics 22(8), 1021–1023 (2006)
2. Aittokallio, B., Schwikowski, B.: Graph-based methods for analyzing networks in cell biology. Briefing in Bioinformatics 7(3), 243–255 (2006)
3. Altaf-Ul-Amin, M., Shinbo, Y., Mihara, K., Kurokawa, K., Kanaya, S.: Development and implementation of an algorithm for detection of protein complexes in large interaction networks. BMC Bioinformatics 7(207) (2006)
4. Arnau, V., Mars, S., Marìn, I.: Iterative cluster analysis of protein interaction data. Bioinformatics 21(3), 364–378 (2005)
5. Asburner, S., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., et al.: Gene ontology: tool for the unification of biology. the gene ontology consortium. Nature Genetics 25, 25–29 (2000)
6. Bader, G., Hogue, H.: An automated method for finding molecular complexes in large protein-protein interaction networks. BMC Bioinformatics 4(2) (2003)
7. Blatt, M., Wiseman, S., Domany, E.: Superparamagnetic clustering of data. Phisical Review Letters 76(18), 3251–3254 (1996)

8. Brohèe, S., van Helden, J.: Evaluation of clustering algorithms for protein-protein interaction networks. BMC Bioinformatics 7, 488 (2006)
9. Ceol, A., et al.: Mint, the molecular interaction database: 2009 update. Nucleic Acids Research 38(Database issue), D532–D539 (2010)
10. Cho, Y.-R., Hwang, W., Zhang, A.: Identification of overlapping functional modules in protein interaction networks: Information flow-based approach. In: Proc. of the Sixth Int. Conf. on Data Mining-Workshops, ICDMW 2006 (2006)
11. Chua, H.N., Ning, K., Sung, W.K., Leong, H.W., Wong, L.: Using indirect protein-protein interactions for protein complex prediction. In: Proceedings of Computational Systems Bioinformatics Conference (CSB 2007), pp. 97–109 (2007)
12. Derenyi, I., Palla, G., Vicsek, T.: Clique percolation in random networks. Physical Review Letters 94(16), 160–202 (2005)
13. Enright, A.J., Dongen, S.V., Ouzounis, C.A.: An efficient algorithm for large-scale detection of protein families. Nucleic Acids Research 30(7), 1575–1584 (2002)
14. Farutin, V., Robinson, K., Lightcap, E., Dancik, V., Ruttenberg, A., Letovsky, S., Pradines, J.: Edge-count probabilities for the identification of local protein communities and their organization. Proteins: Structure, Function, and Bioinformatics 62, 800–818 (2006)
15. Fortunato, S.: Community detection in graphs. Physics Reports 486, 75–174 (2010)
16. Georgii, E., Dietmann, S., Uno, T., Pagel, P., Tsuda, K.: Enumeration of condition-dependent dense modules in protein interaction networks. Bioinformatics 25(7), 933–940 (2009)
17. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. Proc. National. Academy of Science USA 99, 7821–7826 (2002)
18. Hartwell, L.H., Hopfield, J.J., Leibler, S., Murray, A.W.: Clustering algorithm based graph connectivity. Nature 402, C47–C52 (1999)
19. Hwang, W., Cho, Y.-R., Zhang, A., Ramanathan, M.: A novel functional module detection algorithm for protein-protein interaction networks. Algorithms for Molecular Biology 1(24) (2006)
20. Jain, R.D.A.: Algorithms for Clustering Data. Prentice-Hall (1988)
21. Jancura, P., Marchiori, E.: Detecting high quality complexes in a PPI network by edge deletion and node expansion. In: CIBB (2011)
22. Jancura, P., Mavridou, E., Carrillo-De Santa Pau, E., Marchiori, E.: A methodology for detecting the orthology signal in a ppi network at a functional complex level. BMC Bioinformatics (2011) (accepted for publication)
23. Jiang, P., Singh, M.: SPICi: a fast clustering algorithm for large biological networks. Bioinformatics 26(8), 1105–1111 (2010)
24. King, A.D., Przulj, N., Jurisica, I.: Protein complex prediction via cost-based clustering. Bioinformatics 20(17), 3013–3020 (2004)
25. Kovacs, I.A., Palotai, R., Szalay, M.S., Csermely, P.: Community landscapes: an integrative approach to determine overlapping network module hierarchy, identify key nodes and predict network dynamics. PLoS ONE 5(9) (2010)
26. Li, X.L., Foo, C.S., Ng, S.K.: Discovering protein complexes in dense reliable neighborhoods of protein interaction networks. In: Proceedings of Computational Systems Bioinformatics Conference (CSB 2007), pp. 157–168 (2007)
27. Li, X.L., Wu, M., Kwoh, C.K., Ng, S.K.: Computational approaches for detecting protein complexes from protein interaction network: a survey. BMC Bioinformatics 9 (2010)
28. Lin, C., Cho, Y., Hwang, W., Pei, P., Zhang, A.: Clustering methods in protein-protein interaction network. In: Knowledge Discovery in Bioinformatics: Techniques, Methods and Application. John Wiley & Sons, Inc., (2006)

29. Liu, H., Liu, J.: Clustering Protein Interaction Data Through Chaotic Genetic Algorithm. In: Wang, T.-D., Li, X., Chen, S.-H., Wang, X., Abbass, H.A., Iba, H., Chen, G.-L., Yao, X. (eds.) SEAL 2006. LNCS, vol. 4247, pp. 858–864. Springer, Heidelberg (2006)
30. Lubovac, Z., Gamalielsson, J., Olsson, B.: Combining functional and topological properties to identify core modules in protein interaction networks. Proteins: Structure, Function, and Bioinformatics 64, 948–959 (2006)
31. Macropol, K., Can, T., Singh, A.: Rrw: repeated random walks on genome-scale protein networks for local cluster discovery. BMC Bioinformatics 10(1), 283 (2009)
32. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: A survey. IEEE Trans. on Comp. Biol. and Bioinf. 1(1), 24–45 (2004)
33. Palla, G., Derenyi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature 435, 814–818 (2005)
34. Pavlopoulos, G.A., Secrier, M., Moschopoulos, C.N., Soldatos, T.G., Kossida, S., Aert, J., Schneider, R., Bagos, P.G.: Using graph theory to analyze biological networks. BioData Mining 4(10) (2011)
35. Pei, P., Zhang, A.: A two-step approach for clustering proteins based on protein interaction profiles. In: IEEE Int. Symposium on Bioinformatics and Bioengeneering (BIBE 2005), pp. 201–209 (2005)
36. Pereira, J.B., Enright, A.J., Ouzounis, C.A.: Detection of functional modules from protein interaction networks. Proteins: Structure, Fuctions, and Bioinformatics (20), 49–57 (2004)
37. Pizzuti, C.: GA-Net: A Genetic Algorithm for Community Detection in Social Networks. In: Rudolph, G., Jansen, T., Lucas, S., Poloni, C., Beume, N. (eds.) PPSN 2008. LNCS, vol. 5199, pp. 1081–1090. Springer, Heidelberg (2008)
38. Pizzuti, C., Rombo, S.E.: PINCoC: A Co-clustering Based Approach to Analyze Protein-Protein Interaction Networks. In: Yin, H., Tino, P., Corchado, E., Byrne, W., Yao, X. (eds.) IDEAL 2007. LNCS, vol. 4881, pp. 821–830. Springer, Heidelberg (2007)
39. Pizzuti, C., Rombo, S.E.: Multi-functional protein clustering in ppi networks. In: Proc. of the 2nd Int. Conf. on Bioinf. Res. and Dev. (BIRD 2008), pp. 318–330 (2008)
40. Pizzuti, C., Rombo, S.E.: Discovering Protein Complexes in Protein Interaction Networks in Biological Data Mining in Protein Interaction Networks. In: Li, X.-L., Ng, S.-K. (eds.) IGI Global- Medical Inf. Science Ref. (2009)
41. Przulj, N.: Functional topology in a network of protein interactions. In: Jurisica, I., Wigle, D. (eds.) Knowledge Discovery in Proteomics. CRC Press (2005)
42. Ravaee, H., Masoudi-Nejad, A., Omidi, S., Moeini, A.: Improved immune genetic algorithm for clustering protein-protein interaction network. In: Proceedings of the 2010 IEEE International Conference on Bioinformatics and Bioengineering, BIBE 2010, pp. 174–179. IEEE Computer Society (2010)
43. Rives, A.W., Galitski, T.: Modular organization of cellular networks. Proc. of the National Academy of Science 100(3), 1128–1133 (2003)
44. Ruan, J., Zhang, W.: Identifying network communities with a high resolution. Physical Review E 77(1) (January 2008)
45. Samantha, M.P., Liang, S.: Predicting protein functions from redundancies in large-scale protein interaction networks. Proc. of the National Academy of Science 100(22), 12579–12583 (2003)

46. Tasgin, M., Bingol, H.: Community detection in complex networks using genetic algorithm. arXiv:0711.0491, 2007 (2007)
47. Tornw, S., Mewes, H.W.: Functional modules by relating protein interaction networks and gene expression. Nucleic Acids Research 31(21), 6283–6289 (2003)
48. von Mering, D., Krause, C., et al.: Comparative assessment of a large-scale data sets of protein-protein interactions. Nature 31, 399–403 (2002)
49. Wang, J., Li, M., Deng, Y., Pan, Y.: Recent advances in clustering methods for protein interaction networks. BMC Genomics 11(S10) (2010)
50. Zhang, S., Chen, H., Liu, K., Sun, Z.: Inferring protein function by domain context similarities in protein-protein interaction networks. BMC Bioinformatics 10, 395 (2009)
51. Zhao, Y., Levina, E., Zhu, J.: Community extraction for social networks. Proceedings of the National Academy of Sciences 108(18), 7321–7326 (2011)

Short-Range Interactions and Decision Tree-Based Protein Contact Map Predictor

Cosme E. Santiesteban-Toca[1,⋆], Gualberto Asencio-Cortés[2], Alfonso E. Márquez-Chamorro[2], and Jesús S. Aguilar-Ruiz[2]

[1] Centro de Bioplantas, University of Ciego de Ávila, Cuba
cosme@bioplantas.cu
[2] University of Pablo de Olavide, Sevilla, Spain
aguilar@upo.es

Abstract. In this paper, we focus on protein contact map prediction, one of the most important intermediate steps of the protein folding problem. The objective of this research is to know how short-range interactions can contribute to a system based on decision trees to learn about the correlation among the covalent structures of a protein residues. We propose a solution to predict protein contact maps that combines the use of decision trees with a new input codification for short-range interactions. The method's performance was very satisfactory, improving the accuracy instead using all information of the protein sequence. For a globulin data set the method can predict contacts with a maximal accuracy of 43%. The presented predictive model illustrates that short-range interactions play the predominant role in determining protein structure.

Keywords: Protein structure prediction, protein contact map prediction, short-range interactions, decision trees.

1 Introduction

The protein structure prediction still being one of the greatest challenges of bioinformatics [1]. And, inter-residual contact maps is a critical step for the inter-residue contacts prediction problem. The ability to make successful predictions involves understanding the relationship between a sequence and its protein structure [2,3,4,5].

Multiple methods to predict contact maps have been developed. Based on *ab initio* approaches, in homology methods, fold recognition, template-based methods, machine learning, neural network and others [6,7,8,9,10,11,12,13,14]. The prediction quality of these methods has not been improved to satisfactory levels, despite of years of attempts. The main reason for this is perhaps that, it is hard to learn long-range dependencies on contact maps, hence it is especially difficult to predict contacts between residues that have large sequence separations. In addition, another important drawback of these methods is the insufficient capacity to explain their knowledge model for the protein's folding process understanding.

The traditional or *ab initio* folding method employs the principle of predicting protein structure from its known amino acid sequence $(a_0, a_1, \ldots, a_n)$, in

⋆ Corresponding author.

M. Giacobini, L. Vanneschi, and W.S. Bush (Eds.): EvoBIO 2012, LNCS 7246, pp. 224–233, 2012.

order to derive the 3D structure of proteins. We know that a protein chain folds spontaneously and leads to a unique three dimensional structure when placed in aqueous solution. The folding process cannot occur by random conformational search for the lowest energy state. Proteins must form the structure in a time-ordered sequence of events, now called a "pathway". The nature of these events, whether they are restricted to "native contacts" (defined as contacts that are retained in the final structure) or whether they might include non-specific interactions, such as a general collapse in size at the very beginning, were left unanswered [15].

In this paper we propose a solution to predict protein contact maps based on short-range interactions. Despite of some evidences of long-range interactions in stabilizing protein folding, the objective of this research is to know how short-range interactions can contribute to a system based on decision trees to learn the correlation among the covalent structures of a protein residues. Taking into account the high degree of flexibility and the simplicity of understanding of a solution based on decision trees, the proposed algorithm employs the Quinlan C4.5 method, according to previous papers [16,17].

This article is structured as follows. A methodology section, which explains the proteins data set selection criteria, the definition of contact maps, the proposed model architecture and the measures employed for the algorithm effectiveness. A results section, we show tabular and graphical experimentation results. Finally, the conclusions of this research.

2 Materials and Methods

2.1 Data Bases

To analyse the effect of short-range interactions on prediction, we use a set of non-homologous proteins of solved 3D structure. Initially, the set counts 2485 proteins with the lowest possible homology (less than 25% of identity), extracted from the Protein Data Bank (PDB) using PDB_select tool. This set is firstly reduced by excluding those proteins which has non-standard amino acid residues. They were excluded those chains whose backbone was broken. They were chosen only the chains whose: structure does not contain redundant sequences; without ligands, to eliminate false contacts due to the presence of hetero-atoms; and, those proteins that do not belong to the same family or have a common origin. Reducing the list to 173 proteins. This data set combines maximum coverage with minimum redundancy following the Fariselli criteria [6].

With the goal of comparing the proposed predictor with previous methods in the state of the art we employed 53 globulin protein sequences proposed by Zhang [2]. This is a set with a few homologous sequences extracted from PDB.

2.2 Contact Maps Definition

Contact maps are compactly 2D representation of 3D conformation of a protein in a symmetrical square matrix of pairwise inter-residue contacts. The calculation of the distances among the residues is determined by Euclidean distance.

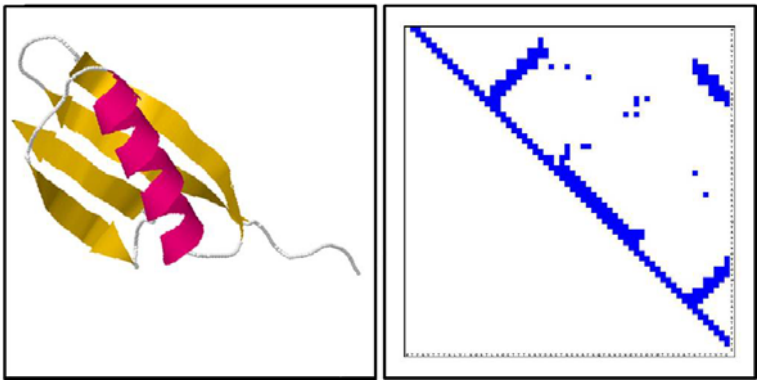


Fig. 1. Contact Map of 2igd protein, constructed with a threshold of 8Å. Left: 3D structure for protein. Right: its contact map showing parallel (top right cluster) and anti parallel sheets (top left and bottom right cluster), and helix features (thin cluster close to main diagonal).

The contact map of a protein (figure 1) is a particularly useful representation of protein structure. This representation provides useful information about the protein's structural motives and it also captures non-local interactions giving clues to its tertiary structure.

2.3 Model Architecture

Decision trees have been proved to be a successful method for prediction of contact maps of proteins [16,17]. Those classifiers make it possible to have understandable rules, which can be used to find further explanations of the data that are classified.

To predict contact map, we use an algorithm based on the Quinlan C4.5 decision tree [18], using the default setting. Our method builds decision trees for all possibles pairs of contacts, which has a total of 400 trees (20 x 20 amino acids). The prediction is treated as a classification problem, which takes into account the contacts or non-contacts between residues.

As input coding, the proposed method introduces the use of short-range interactions as a basis for training the predictor. Taking into account that oligopeptides are a few amino acids covalently joined (up to 10) and the average length of structural motives regions (up to 21), the algorithm employs vectors of length 21. This is equivalent to shift a window of length 21 by the amino acids chain. The built vector includes information of the substring formed among non adjacent amino acids. It is created a vector for each possible short-range interactions that can be formed in the protein (figure 2).

For a couple of amino A_1 A_2, the first 20 elements of the vector match the existing amino acids and contain their frequencies in the substring that is formed

Substring between the pair of amino acids				Class
A_1	A_2	...	A_{20}	Contact

Fig. 2. Scheme of input coding for decision trees. The first 20 bits in the coding, represent the frequency that appears the amino acids in the sub-chain. Where cero means that this amino acid is not present in the sub-chain. The last bit encodes by class (Contact or Not-Contact).

between the pair of amino acids analysed. To define the Class we adopt a threshold value of 8Å.

The decision tree-based predictor of protein contact maps (DTP) is shown in Figure 3. Given the distance matrix of a protein set with known structure (P_1, P_2, ..., P_n), the DTP builds a model of two-dimensional array of size NxN, where N is the number of amino acids (20). Each matrix cell contains a function $f_{(A_1,A_2,S)}$ formed by a decision tree, whose input vector is composed by the amino acids couple (A_1, A_2) and the information extracted from the substring (S) contained between them. For an unknown sequence (S?), each couples of amino acids is evaluated in the built model. The result of prediction is obtained by the occurrence of contact or non-contact.

2.4 The Pre-processing Procedure

Contact map prediction is an unbalanced problem. These maps contain, as average, a number of contacts (N_C) considerably lower than the number of non-contacts (N_{NC}) about 1/13. N_C increases almost linearly with protein sequence length (data not shown). For this reason N_{NC} increases with the square of the protein length.

The C4.5 decision trees. This algorithm is based on the data frequency and it is highly susceptible to the unbalance problem. To avoid the unbalanced effects we edit the data base applying an oversampling method. This method reproduces

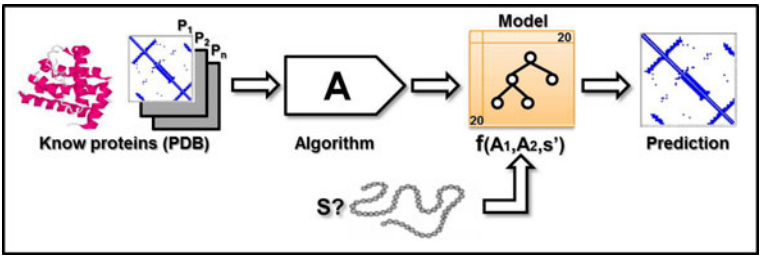


Fig. 3. Scheme of the decision tree-based predictor of protein contact maps. Where P_1 to P_n are the training proteins, A is the algorithm that creates the knowledge model and S? is the unknown sequence.

the minority class until mitigate the problem, taking into account the unbalance-ratio. This value is statistically calculated for each couple of amino acids in the protein. As result, the number of predicted contacts of a residue becomes a function of its structural environment.

2.5 Evaluation of the Efficiency

The effectiveness of prediction (Ap) is calculated as the ratio of true positives (1). This is because this equation penalizes non-contacts and prioritizes contacts.

$$Ap = TP/(TP + FN) \tag{1}$$

In order to compare the effectiveness of the predictor, an extra measure is applied: the improvement over a random predictor (2). This measure computes the ratio between Ap (1) and the accuracy of a random predictor (Nc / Np):

$$R = Ap/(Nc/Np) \tag{2}$$

where Nc is the number of real contacts in the protein of length Lp, and Np are all the possible contacts. In this paper in order to limit the prediction of local contacts (clustered along the main diagonal of the contact map) the proposed procedure does not include contacts between residues whose sequence separation is less than four residues.

3 Results

To study the influence of short-range interactions in the proteins, are analysed the distribution of protein contacts and structural motives with respect to the length of the sequence separation. We used the set of 173 proteins grouped into four classes, according to their sequences length (Ls): Ls $<$ 100 (65 proteins), 100 $\leq$ Ls $<$ 170 (57), 170 $\leq$ Ls $<$ 300 (30) and Ls $>$ 300 (21).

At first, with the aim of the study the distribution of inter-residual contacts, were analysed the frequencies of their appearance depending on the residues separation in the sequence (figure 4). It was used a thresholds range from 5Å to 12Å. It is obvious that most of contacts are concentrated in low sequences separation. Assuming a loss of 5% of contacts, the 95% is concentrated in sequence separations $\leq$ 150 and the 70% are concentrated just in residues with separation 10.

Another interesting analysis is to take into account the length by structural motives regions (helical and beta regions). We also studied the distribution of the number of residues per helical segment and per β-sheet segment (figure 5).

The fact that the length of β-regions in proteins is shorter than the helical segments is clearly shown in figure 5. Helical segment appears in regions from 3 to 20 amino acids and β-segment appears in regions from 2 to 10 amino acids. In average, the 80% of structural motives appears to be in the range of 2 to 10 amino acids.

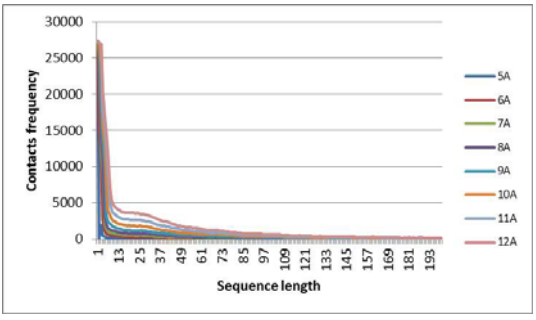


Fig. 4. Contacts distribution histogram. Plotting the contacts frequency as a function of sequence separation, for thresholds of 5Å to 12Å.

The distribution of contacts and structural motives, indicates that contacts in proteins are not randomly distributed and occur, predominantly, among residues with a low sequence separation.

To solve our specific problem, three methods are implemented:

- **DTP**: employs all information included in the protein sequences. The length of sub-sequences is not limited.
- **DTPsi**: method variation that employs as input coding only the short-interactions present. The input vector will be formed by the information of amino acids with maximal sequence separation up to 20.
- **DTPsi_ed**: it is the DTPsi method but we apply a pre-processing algorithm to the input data. Taking into account the unbalanced nature of present classes in this problem, we used an oversampling method to balance the database.

The implemented methods are tested on the selected database using a 10 folds cross-validation procedure. With the intention of highlighting the relationship

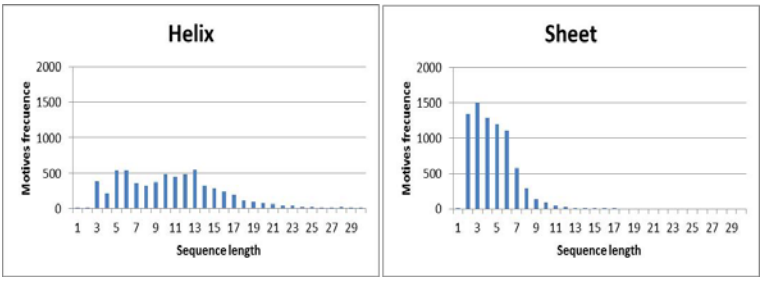


Fig. 5. Distribution of the number of residues per helical segment and per β-sheet segment

Table 1. Comparison of the performance of the different methods used to predict contact maps

	$Ls < 100_{(65)}$		$100 \leq Ls < 170_{(57)}$		$170 \leq Ls < 300_{(30)}$		$Ls \geq 300_{(21)}$	
	Ap	R	Ap	R	Ap	R	Ap	R
DTP	0,12	2,33	0,05	2,75	0,03	4,26	0,03	7,52
DTPsi	0,13	2,10	0,07	2,14	0,06	1,18	0,04	3,47
DTPsi_ed	0,18	1,71	0,14	1,61	0,13	1,38	0,12	1,49

between the results and the proteins size, the values of effectiveness were calculated after grouping proteins according to their sequence length (table 1).

The results show that, in general, for all proteins, the algorithm trained with short-range interactions (DTPsi) show a good behaviour. DTPsi not only improves the minimum efficiency threshold proposed by the DTP algorithm, when is applied an algorithm to balance the class (DTPsi_ed), it improves drastically the prediction effectiveness.

Figure 7 shows the effectiveness of predictions based on the proteins length, using different methods (DTP, DTPsi and DTPsi_ed). This graph shows that the effectiveness of the algorithm is dependent on the length of the protein. However, like the rest of algorithms, DTPsi_ed is more efficient to predict contacts in short sequences and it's efficiency decreases when the sequence length is incremented.

3.1 Comparison with the Previous Methods

To compare the accuracy of our algorithm with respect to the previous methods we used the set of 53 proteins. Here the protein sequences are grouped into four classes: Ls<100, 100≤Ls<200, 200≤Ls<300, Ls>300, according to their sequences' length (Ls). The proteins 1TTF, 1E88, 1NAR, 1BTJ_B and 1J7E_A, were used to test the trained algorithm. The proposed procedure does not include contacts between residues whose sequence separation is less than four, to avoid small ranges of false contacts.

The table 2 shows the comparative results for the algorithms: Occ (Occupancy method) [19], based on a filtered procedure, reached an accuracy about 26%; Net_75 method [20], it uses multiple sequence alignment as input for a classical feed-forward neural network trained with a standard back-propagation algorithm, reached the accuracy of about 28%; RBFNN method [2] uses a binary input encoding scheme with a radial-based function neural network optimized by a genetic algorithm, reached an accuracy of 32%; and DTPsi_ed, achieved the best accuracy: 43%.

Considering the relationship between the residue length and the average accuracy, our algorithm can improve the prediction performance dramatically. Except for sequence length less than 100 where there are not differences respect to RBFNN method.

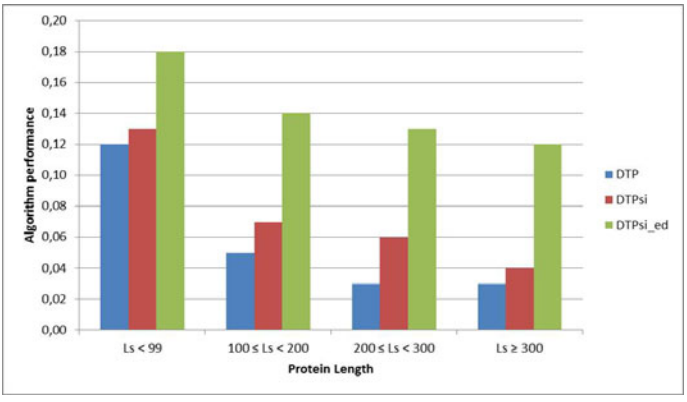


Fig. 6. This graph shows the efficiency of the contacts prediction based on the sequence lengths of proteins. In the x-axis values are represented the effectiveness achieved by the predictors, depending on the length of the sequences. The vertical axis represents the effectiveness values.

Table 2. Comparison of the predictors accuracy: Occ, Net_75, RBFNN and DTPsi_ed (our method). Ls is the length of the protein sequence. For this comparison it was employed the experimental results reported by Zhan[2].

Methods	Ls<100	100≤Ls<200	200≤Ls<300	Ls>300
Occ	0,26	0,21	0,15	0,10
Net75	0,26	0,28	0,21	0,20
RBFNN	0,30	0,31	0,32	0,28
DTPsi_ed	0,30	0,43	0,35	0,29

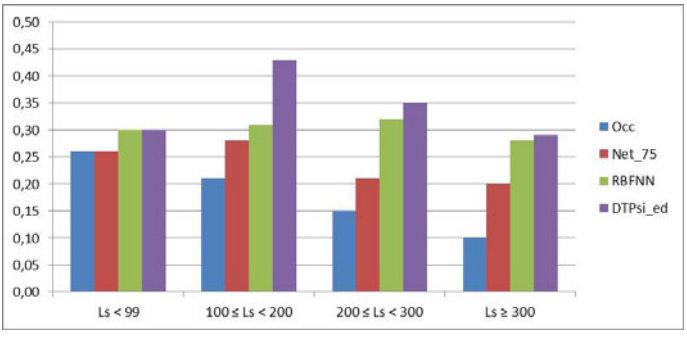


Fig. 7. This graph shows the comparative results in the prediction of contacts considering the sequence lengths of proteins. In the x-axis are represented the values effectively achieved by the predictors, depending on the length of the sequences. The vertical axis represents the effectiveness.

4 Conclusions

The presented predictive model illustrates how short-range interactions play a predominant role in determining protein structure. The proposed method combines the use of decision trees with a new input encoding for short-range interactions. The method performance was very satisfactory. It improves the accuracy with respect to the obtained by the DTP method. In a comparison with reported algorithm for a globulin data set, DTPsi_ed can predict contacts with a maximal accuracy of 43%.

Acknowledgements. This research is inserted in the doctoral program in Soft Computing, developed by the University of Las Villas in Cuba and the Andalusian Universities, under the sponsorship of the AUIP, which has promoted and apported the financial support to the entire program and research visits. Special thanks to Lic Nataniel Giménez Velázquez and Ernesto Estrada Cruz by their contributions.

References

1. Ouzounis, C.A., Valencia, A.: Early bioinformatics: the birth of a discipline a personal view. Bioinformatics 19(17), 2176–2190 (2003)
2. Quan, Z.H., Zhang, G.-Z., Huang, D.S.: Combining a binary input encoding scheme with RBFNN for globulin protein inter-residue contact map prediction. Pattern Recognition Letters 26, 1543–1553 (2005)
3. Glasgow, J., Kuo, T., Davies, J.: Protein structure from contact maps: A case-based reasoning approach. Inf. Sys. Front 8, 29–36 (2006)
4. Ramanathan, A.: Using Tensor Analysis to characterize Contact-map Dynamics of Proteins. PhD thesis, Carnegie Mellon University Pittsburgh, PA (2008)
5. Zhou, J., Arndt, D., Wishart, D.S., Lin, G., Shi, Y., Zhou, J., Arndt, D., Wishart, D.S., Lin, G.: Protein contact order prediction from primari sequences. BMC Bioinformatics 9(255), 1–21 (2008)
6. Fariselli, P., Olmea, O., Valencia, A., Casadio, R.: Prediction of contact maps with neural networks and correlated mutations. Protein Engineering 14(11), 835–843 (2001)
7. Pollastri, G., Baldi, P.: Prediction of contact maps by GIOHMMs and recurrent neural networks using lateral propagation from all four cardinal corners. Bioinformatics 18, 1–9 (2002)
8. Kim, H.: Computational analysis of hydrogen bonds in protein-RNA complexes for interaction patterns. FEBS Letters 552, 231–239 (2003)
9. Martin, A.J.M., Walsh, I., Bau, D.: Ab initio and template-based prediction of multi-class distance maps by two-dimensional recursive neural networks. BMC Structural Biology 9(5), 1–38 (2009)
10. Ahmad, M., Mathkour, H.: An integrated approach for protein structure prediction using artificial neural network. In: 2010 Second International Conference on Computer Engineering and Applications, pp. 484–488. IEEE (2010)
11. Sinha, S., Durga Bhavani, S., Suvarnavani, K.: Mining of protein contact maps for protein fold prediction. WIREs Data Mining Knowl. Discov. 1(4), 362–368 (2011)

12. Saraee, M., Korbekandi, H., Habibi, N.: Protein contact map prediction using committee machine approach. International Journal of Data Mining and Bioinformatics 2, 205–209 (2011)
13. Hossein, M., Narjes, S., Habibi, K.: Protein contact map prediction based on an ensemble learning method. In: 2009 International Conference on Computer Engineering and Technology 2009, vol. 2, pp. 205–209. IEEE (2009)
14. Min, H., Yoon, S., Kim, J., Kim, H.: Constructing accurate contact maps for hydroxyl-radical-cleavage-based high-throughput rna structure inference. IEEE Transactions on Biomedical Engineering 58(5), 1347–1355 (2011)
15. Shao, Y., Bystroff, C., Zaki, M.J., Hu, J., Shen, X.: Mining Protein Contact Maps. In: BIOKDD 2002: Workshop on Data Mining in Bioinformatics (with SIGKDD 2002 Conference), pp. 3–10 (2002)
16. Toca, C.E.S., Márquez Chamorro, A.E., Asencio Cortes, G., Aguilar Ruiz, J.S.: A Decision Tree-Based Method for Protein Contact Map Prediction. In: Giacobini, M. (ed.) EvoBIO 2011. LNCS, vol. 6623, pp. 153–158. Springer, Heidelberg (2011)
17. Santiesteban-Toca, C.E., Aguilar-Ruiz, J.S.: DTP: Decision Tree-Based Predictor of Protein Contact Map. In: Mehrotra, K.G., Mohan, C.K., Oh, J.C., Varshney, P.K., Ali, M. (eds.) IEA/AIE 2011, Part II. LNCS, vol. 6704, pp. 367–375. Springer, Heidelberg (2011)
18. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann (1993)
19. Valencia, A., Olmea, O.: Improving contact predictions by the combination of correlated mutations and other sources of sequence information. Protein Engineering 2, S25–S32 (1997)
20. Casadio, R., Fariselli, P.: A neural network based predictor of residue contacts in proteins. Protein Engineering 12(1), 15–21 (1999)

A NSGA-II Algorithm for the Residue-Residue Contact Prediction

Alfonso E. Márquez-Chamorro[1], Federico Divina[1], Jesús S. Aguilar-Ruiz[1], Jaume Bacardit[2], Gualberto Asencio-Cortés[1], and Cosme E. Santiesteban-Toca[3]

[1] School of Engineering, Pablo de Olavide University of Sevilla, Spain
{amarcha,fdivina,aguilar,guaasecor}@upo.es
[2] School of Computer Science, University of Nottingham, United Kingdom
jaume.bacardit@nottingham.ac.uk
[3] Centro de Bioplantas, University of Ciego de Avila, Cuba
cosme@bioplantas.cu

Abstract. We present a multi-objective evolutionary approach to predict protein contact maps. The algorithm provides a set of rules, inferring whether there is contact between a pair of residues or not. Such rules are based on a set of specific amino acid properties. These properties determine the particular features of each amino acid represented in the rules. In order to test the validity of our proposal, we have compared results obtained by our method with results obtained by other classification methods. The algorithm shows better accuracy and coverage rates than other contact map predictor algorithms. A statistical analysis of the resulting rules was also performed in order to extract conclusions of the protein folding problem.

Keywords: Protein structure prediction, contact map, multi-objective evolutionary computation.

1 Introduction

Protein Structure Prediction (PSP) is one of main challenges in Structural Bioinformatics. Since Anfinsen's experiment discovered that the amino acid sequence determines the shape of a protein [1], a huge number of computational experiments were performed with the aim of obtaining the rules of the protein folding. Knowledge of these rules would play an important role in Biomedicine for the design of new drugs. Although experimental procedures to obtain the 3D protein structure, as X-ray Crystallography and Nuclear Magnetic Resonance (NMR), have shown brilliant results [2], the cost of such techniques, both in term of time and money, is prohibitive. Besides, these techniques cannot be applied to all proteins. In fact, 25% of proteins do not crystallize and are too big for the NMR.

For these reasons, computational methods are particularly suited for this problem, since they, generally, represent a cheaper and faster way to address the

M. Giacobini, L. Vanneschi, and W.S. Bush (Eds.): EvoBIO 2012, LNCS 7246, pp. 234–244, 2012.

protein folding problem. Some of these computational methods used a contact map representation to solve this problem. A contact map is a bi-dimensional representation of the protein structure of a protein, where if an entry i, j has value 1 then a contact between residues i and j is predicted, and a 0 indicates a no contact. We consider a contact between i and j, if the distance between them is lower than a certain threshold μ. Different approaches were developed as protein contact map predictors: artificial neural networks (ANNs) [3,4], support vector machines [5], evolutionary algorithms (EAs) [6] and template-based modeling [7]. Every two years, Critical Assessment of Protein Structure Prediction (CASP) competition [8] evaluates the most accurate computational methods for the PSP problem. One of the categories of this competition is called "Detecting residue-residue contacts in proteins (RR)". Our approach is included in this category.

Among the above mentioned methods, EAs, have become popular as robust and effective methods for solving optimization problems. In particular, they have shown the capacity of finding suboptimal solutions in search spaces when the search space is characterized by high dimensionality. This is the case for the protein folding problem, where the set of possible folding rules of a protein determine the search space. Many evolutionary approaches have been developed to tackle the PSP problem, e.g., [9] [10] [6] [11]. These methods evaluate individuals by means of a single function that provides a measure of their quality. In other words, they are evaluating a single objective function. This approach represents the classical way of addressing a problem with an EA: the objectives to optimize are combined into a single fitness function which is then used in order to guide the evolutionary search. However, there are some problem where this approach is not the most appropriate. Different solutions can produce conflicts between different objectives. A solution that is optimal with respect to one objective may not be optimal for the rest, therefore it would be improper to choose such solution as optimal solution of the problem. It becomes then necessary to establish a compromise among the objectives. The solutions that fulfill this compromise are called the Pareto set. The notion of Pareto set is based on the concept of dominance that will be explained in the next section. When an optimization problem has several objectives, the task of finding one or more suboptimal solutions is called Multi-objective optimization.

Multi-objective Evolutionary Algorithms (MOEAs) appear as an extension of EAs for single objective problems. A MOEA should be designed to achieve two purposes simultaneously: to achieve good approximations to the Pareto front and maintain the diversity of solutions, in order to adequately search the solution space and do not converge to a unique solution [12]. Some of the best known MOEAs are NSGA, SPEA, NSGA-II, SPEA-II and PAES-II [8].

Several prediction methods have considered the PSP problem as a multi-objective optimization problem. For instance, [13] developed MI-PAES as a modified version of PAES using a torsion angles model. A parallel multi-objective optimization was performed by using Chemistry at HARvard Macromolecular Mechanics (CHARMM) energy function in [14]. [15] proposed a multi-objective Feature Analysis and Selection Algorithm (MOFASA) in order to solve the

Protein Fold Recognition (PFR) problem. In [16], a I-PAES algorithm is used as search procedure for exploring the space of the PSP problem. The concept of bond and non-bond energies are included in the fitness function of this approach.

In this paper, we propose a contact map predictor based on a MOEA. More specifically, it is based on a NSGA-II algorithm [17]. A NSGA-II algorithm initially creates a population (random or by a technique of initialization) of parents. The population is sorted according to levels of non-dominance (ranking Pareto fronts). Each solution is then assigned a fitness value according to their level of non-dominance (1 is the best level). Tournament selection, the crossover and mutation are used to create the offspring population of size N.

Our algorithm generates a set of rules that predicts contacts between amino acids. In particular, each rule imposes a set of conditions on some specific amino acids properties. Rules consider two windows of 3 amino acids, which are centered around the two target residues in contact.

In order to test our proposal, we obtain the training data set from the Protein Data Bank (PDB), and produce a file in arff format with the encoded information. The rules that are produced after the training phase are classified according to each specific pair of residues that they represents. For a new protein sequence, we apply the required rules for each residue pair and obtain the protein contact map. Our application also provide a graphical representation of these contact maps. The novelty of our proposal consists on the use of amino acid properties which are involved in the folding process and, to the best of our knowledge, have not been applied in similar evolutionary approaches for this problem.

The remainder of this paper is organized as follows. Section 2 introduces some basic concepts of the Multi-objective optimization. Our multi-objective evolutionary approach is described in section 3. Section 4 presents the experimentation and obtained results. Finally, section 5, includes some conclusions and possible future works.

2 Multi-objective Optimization Problem

Before describing our algorithm, this section presents a brief introduction to multi-objective optimization problems and related concepts.

A Multi-objective optimization problem is based on the optimization (minimization or maximization) of a set of objective functions, usually in conflict with each other. The existence of multiple objective poses a fundamental difference with the single objective problems: typically there will not be a single solution, but a set of solutions that can present different clashes among the values of the objectives to optimize. We can define a multi-objective optimization problem in this way: let $(f_1(x), f_2(x)...f_n(x))$ be a set of functions to be optimized, where $x = (x_1, ..., x_p)$ is a vector of decision variables belonging to a universe X and $f_i(x)$ is an arbitrary linear or non-linear function, $1 \leq i \leq n$. Therefore, the problem consists of finding the x that provides the best compromise value for all $f_i(x)$.

To solve the above problem, we should defined some criteria to determine which solutions are considered of good quality and which are not. Hence, we introduce the concept of dominance, that is used in the process of evaluating the different solutions. A solution x is said to be not dominated *iff* there is not another solution y such that: $f_i(y) <= f_i(x)$ for all $i = 1..n$ and $f_i(y) < f_i(x)$ for some i. From this, it follows that the Pareto front is formed by all the non-dominated solutions.

We have applied these concepts to the Protein Contact Prediction problem. In this article, we have considered coverage and accuracy as two different functions and are optimized separately.

In order to do so, we have implemented a MOEA based on an Elitist Non-Dominated Sorting Genetic Algorithm (NSGA-II). NSGA-II incorporates elitism and reduces the complexity of the procedure fast sorting by non-dominance of its predecessor NSGA. The algorithm performs a classification of the population using Pareto fronts. Individuals which belong to the first front are the non-dominated front, those in the second front are not dominated in the absence of previous front, and so on. Each individual is assigned a rank equal to its level of non-dominance. The best individuals are those with lower ranks. In order to maintain diversity, we use a crowding distance, which is assigned to each individual of the current population. The selection is performed by binary tournament. The tournament is won by the individual with a lower range (Pareto front level). If the two ranges are the same, the tournament is won by the individual who has lower crowding distance. This algorithm has a low time complexity of $O(NlogN)$, where N is the population size.

3 Our Approach

In this section, we present the main characteristics of our proposal. As we have said before, the aim of this algorithm, called PSP-NSGAII, is the prediction of protein contact maps. In order to test our proposal, the first thing to do was to select a set of sequences. For this, we selected from PDB a set of 173 proteins that appears in [3]. We extract the required information as the amino acid sequences and distances between amino acids. To calculate the distances, we use the Euclidean distance between C_β atoms (C_α for Glycine) of each pair of residues. The formula of Euclidean distance is $d(i,j) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$, where (x_1, y_1, z_1) represent the atomic coordinates of the first amino acid and (x_2, y_2, z_2) are the coordinates of the second amino acid. Once the training set is prepared, we use this data to train our evolutionary algorithm. As we have said previously, we propose a Multi-Objective algorithm as a method to identify protein folding rules. These rules provide us the specified characteristics of amino acids in contact. They specify which property values and conditions must have the amino acids in contact and the ones which precede and follow them. Our proposal build the set of final rules in an incremental way. Each time the algorithm is run, a set of rules are selected and added to a final solution set. For each iteration, we select those rules which contribute to increase the F-measure of the global solution.

In the following the characteristics of the representation, the fitness function and the genetic operators used by the EA will be presented.

3.1 Encoding

Each individual is represented as follows. We have taken into account six amino acids. For two amino acid in contact i and j, we represent the amino acid $i-1$, $i+1$, $j-1$ and $j+1$, i.e., the amino acids that precede and follow i and j in the sequence. This choice was made after having performed various experiments with different window sizes, ranging from 6 to 14. Each amino acid is represented by 7 genes; two genes for the hydrophobicity (ranging from -1 to 1), two genes for polarity (ranging from -1 to 1), 1 gene for the charge (-1, 0, 1 for negative, neutral and positive charge respectively) and two genes for the volume of residue (ranging from 0 to 1). Figure 1 shows the representation for an amino acid. Our representation consists in 42 attributes in total.

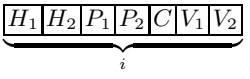


Fig. 1. Example of encoding for the amino acid i. An individual is constituted by six amino acids $i-1$, i, $i+1$, $j-1$, j and $j+1$. H_1,H_2,P_1,P_2,V_1 and V_2 are lower and upper bounds for the hydrophicity, polarity and volume values, respectively. C represents the charge value of the residue.

We selected Kyte-Doolittle hydropathy profile [18], the Grantham's profile [19] for polarity and Klein's scale for net charge [20]. The Dawson's scale [21] is employed to determine the volume of the residues. In table 1, we can appreciate the amino acid values for each property according to the cited scales and normalized between -1 and 1 for hydrophobicity and polarity, and between 0 and 1 for the residue volume.

From all the extracted data, we have built a file in arff format, with all the training data information. This file is available at *http://www.upo.es/eps/asencio/data/training_set.arff*. In this file we include protein subsequences of windows of six amino acids codified with the values of the cited four different physico-chemical properties. The positive class (contact) is represented with 1 and the negative class (no contact) is represented with 0. The total data set constitutes $123,949$ instances with $6,922$ positive and $117,027$ negative cases (contact and no contacts respectively).

3.2 Fitness Function

As already mentioned, we consider two objectives to be optimized: coverage and accuracy. Coverage represents the number of predicted contacts and accuracy evaluates the real predicted contacts rate. Therefore, $Coverage = C/C_t$ and $Accuracy = C/C_p$, where C is the number of correctly predicted contacts of a

Table 1. Values of different properties according to the cited scales for each amino acid. H represents the hydrophobicity, P the polarity, C the charge net and V is the residue volume.

Prop.	A	C	D	E	F	G	H	I	K	L
H	0.40	0.56	-0.78	-0.78	0.62	-0.09	-0.71	1.00	-0.87	0.84
P	-0.21	-0.85	1.00	0.83	-0.93	0.01	0.36	-0.93	0.58	-1.00
C	0	0	-1	-1	0	0	0	0	1	0
V	0.33	0.40	0.33	0.67	0.87	0.07	0.80	0.73	0.93	0.73

Prop.	M	N	P	Q	R	S	T	V	W	Y
H	0.42	-0.78	-0.36	-0.78	-1.00	-0.18	-0.16	0.93	-0.20	-0.30
P	-0.80	0.65	-0.23	0.38	0.38	0.06	-0.09	-0.75	-0.88	-0.68
C	0	0	0	0	1	0	0	0	0	0
V	0.80	0.67	0.73	0.80	1.00	0.40	0.67	0.67	0.93	0.93

protein, C_t is the total number of contacts of the protein and C_p is the number of predicted contacts. We aim at finding the best compromise between these two measures.

3.3 Genetic Operators

We use two mutation operators. The first operator follows a Gaussian distribution for a randomly selected individual and increase or decrease a gene value with a probability of 0.5. A second operator randomly selects a gene that is related to a given property, and moves the bounds to the maximum or minimum of the domain, making the property irrelevant in this rule. For example, if the property is the hydrophobicity, we change the range to -1, 1 so the rule does not take into account this property in this case. This type of mutation is applied with a 0.1 probability. For each individual, we test that the mutated value was between the allowed ranges.

A 2-point crossover operation was used with a 0.5 probability. A binary tournament selection is applied with a probability of 0.5. In each tournament, we select the individual which is located in the better Pareto front. If the two individuals are on the same front, we use the crowding distance to determine the winning configuration. The crowding distance is a measure of the diversity of the population. This process is called Stacking tournament selection.

The population size is set to 100, and the initial population is randomly initialized. The maximum number of generations that can be performed is set to 100. However, if the fitness of the best individual does not increase over twenty generations, the algorithm is stopped and a solution is provided. At the end of the execution, repeated or redundant rules are discarded from the solution set.

All the parameters were set after having performed several trial runs of the algorithm.

4 Experiments and Results

As mentioned in the previous section, in order to test our proposal, we have selected a protein data set specified in [3]. This data set consists of 173 proteins with percentage of sequence identity lower than 25%. Four subsets have been classified according to the sequence length; lower than 100 residues (DS1), between 100 and 170 (DS2), between 170 and 300 (DS3), and higher than 300 residues (DS4). The minimum and maximum size of the proteins are 31 and 753 amino acids respectively. A threshold of 8 angstroms (Å) was established to determine a contact. In order to avoid the effect of learning local contacts, we set a minimum sequence separation of 7 residues between each pair of amino acids to establish a contact. A 3-fold cross-validation were performed during all the experimentations. All these requirements were also found in [3]. In order to validate our experimentations, accuracy and coverage rates were calculated. These two measures are also employed to validate the prediction algorithms in CASP competitions.

We have performed several experiments with three Weka classifiers [22]: Näive Bayes (NB), C4.5 classifier tree (J48), Nearest Neighbor approach with $k = 1$ (IB1). The obtained results can be seen in Table 2 for a 3-fold cross-validation. We appreciate low coverage and accuracy values in all the cases. The training data used contained all the possible subsequences of size 6 of the DS1 protein data set with a minimum separation between contact residues of 7 amino acids. This experiment was performed with the aim of validating our representation and confirms that this representation provides enough information for a good performance of a learning classifier. Moreover, we can also notice that PSP-NSGAII achieved the best results for this experiment.

Table 2. Average results obtained for different classification Weka algorithms for the DS1 protein data set

Algorithm	Data Set	$Coverage_{\mu}$	$Accuracy_{\mu}$
J48	DS1	0.03	0.31
IB1	DS1	0.09	0.09
NB	DS1	0.20	0.13
PSP-NSGAII	DS1	0.21	0.33

The optimal number of rules for the prediction is unknown. In order to establish the optimal number of executions, we have run several preliminary experiments and compared the obtained results. From these, we have concluded that the best results were obtained when the algorithm was run for 1,000 executions.

Table 3 shows the average results obtained using the dataset. Our results were compared with the ones showed in [3]. We can observe as main conclusion, how the coverage and accuracy rates decrease if the size of the proteins increases. This is due to the fact that, generally, *ab initio* methods only work well with peptides lower than 150 amino acids [23]. We obtain better results for proteins whose

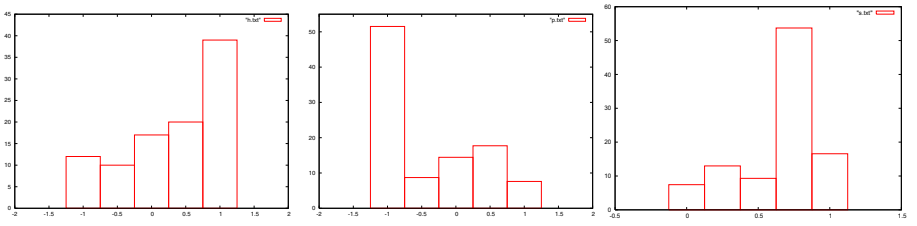
Table 3. Average results and standard deviation obtained for 1,000 executions of the algorithm for the different protein data subsets

Data Set	#proteins	$Coverage_{\mu\pm\sigma}$	$Accuracy_{\mu\pm\sigma}$	$Accuracy_{\mu}$ [3]
DS1	65	$0.21_{\pm 0.02}$	$0.33_{\pm 0.01}$	0.26
DS2	57	$0.10_{\pm 0.01}$	$0.21_{\pm 0.02}$	0.21
DS3	30	$0.08_{\pm 0.03}$	$0.13_{\pm 0.02}$	0.15
DS4	21	$0.06_{\pm 0.03}$	$0.09_{\pm 0.03}$	0.11

sequence length is lower than 100 (DS1), 0.33 against 0.26. We have obtained the same accuracy rate for the second subset DS2, and similar rates for the third and fourth group. We could not compare the coverage rates, because they are not included in the cited paper [3].

We have analyzed the set of resulting rules, and they show that a vast majority of amino acids in contact have high values of hydrophobicity. On the other hand, a high percentage of contacts have non-polar residues. These conclusions were expected, because hydrophobic and non-polar amino acids tend to be located in the inner of the protein. Therefore, these type of residues have more probabilities to be in contact [6]. According to the residue volume, residues with values between 0.5 and 0.75 are the most representative. We have not observed any clear conclusion according to the net charge. Although the amino acids with opposite charges are supposed to be in contact [6], this condition seems to be irrelevant in our rule set and does not appear as a clear conclusion. Figure 2 shows the graphical representation of the probability of appearance of each property in our whole set of resulting rules for the amino acid i. The properties values have been discretized in five groups in intervals of 0.5 from -1 to 1 for the hydrophobicity and polarity and from 0 to 1 in intervals of 0.25 for the residue volume. The rest of amino acid positions in the rules presents similar behaviors.

In figure 3, we show a graph which represents the different Pareto fronts for five generations (from generation 0 to 80 with an interval of 20) of an execution in order to test the correct performance of our multi-objective evolutionary algorithm. Each different symbol represents an individual of the Pareto front in different generations. The X-axis represents the coverage and the Y-axis shows the accuracy rate. These two measures are the two parameters which should be optimized during the executions. We can notice how the quality of individuals improve with the generations.



(a) Hydrophobicity (b) Polarity (c) Residue volume

Fig. 2. Representation of appearance percentages for the different properties at the i-position of the rules

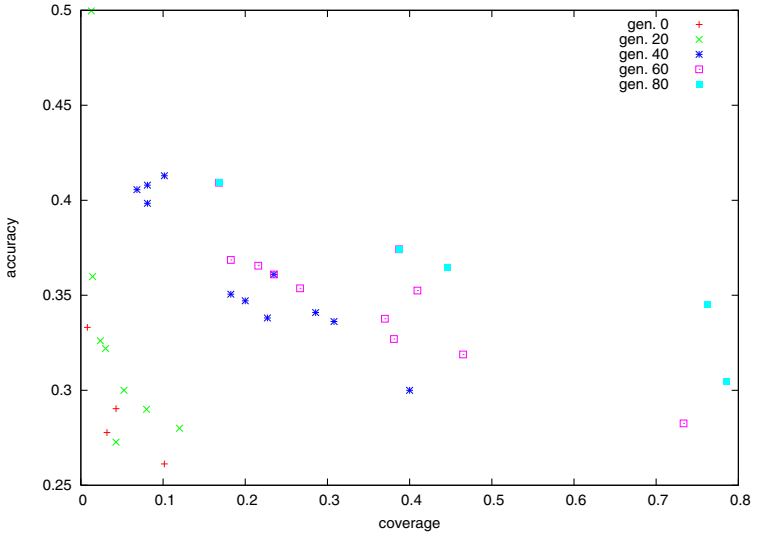


Fig. 3. First Pareto fronts for an execution in different generations

5 Conclusions and Future Work

In this work, we presented a multi-objective optimization algorithm for the residue-residue contact prediction. This algorithm generates rules that predict the necessary conditions for the contact between two amino acids based on their physico-chemical properties. The algorithm was tested on a set of protein sequences that had been previously used in the literature and achieve similar coverage and accuracy rates than other contact map predictor algorithm. We have analyzed the resulting rules set and drawn some conclusions about the folding prediction problem. From the obtained results, we can conclude that our algorithm, as other *ab initio* methods, obtains lower accuracy if the size of the protein is increased. Although these methods are computationally expensive,

they have a main advantage; by only taking the sequence as baseline information, it is possible to obtain a folding model for an unknown protein.

As future work, we are planning to include more useful information based on amino acid properties in our rules representation as secondary structure prediction and solvent accessibility. The variability of the window size must be taken into account for the next version of the algorithm. Furthermore, our algorithm must be validated with a higher number of proteins data set.

Acknowledgements. This research was supported by by the Junta de Andalucia, Project of Excellence P07-TIC-02611 and by Spanish Ministry of Science and Technology under grants TIN2007-68084-C02-00 "Sistemas Inteligentes para descubrir patrones de comportamiento. Aplicación a base de datos biológicas".

References

1. Anfinsen, C.: The formation and stabilization of protein structure. The Biochemical Journal 128, 737–749 (1972)
2. Bashan, A., Yonath, A.: Ribosome crystallography: From early evolution to contemporary medical. Ribosomes Structure, Function, and Dynamics, 3–18 (2011)
3. Fariselli, P., Olmea, O., Valencia, A., Casadio, R.: Prediction of contact map with neural networks and correlated mutations. Protein Engineering 14, 133–154 (2001)
4. Tegge, A.N., Wang, Z., Eickholt, J., Cheng, J.: Nncon: Improved protein contact map prediction using 2d-recursive neural networks. Nucleic Acids Research 37(2), 515–518 (2009)
5. Cheng, J., Baldi, P.: Improved residue contact prediction using support vector machines and a large feature set. Bioinformatics 8, 113 (2007)
6. Gupta, N., Mangal, N., Biswas, S.: Evolution and similarity evaluation of protein structures in contact map space. Proteins: Structure, Function, and Bioinformatics 59, 196–204 (2005)
7. Zhang, Y.: I-tasser: fully automated protein structure prediction in casp8. Proteins: Structure, Function, and Bioinformatics 77, 100–113 (2009)
8. Kinch, L.N., Shi, S., Cheng, H., Qian Cong, Q., Pei, J., Mariani, V., Schwede, T., Grishin, N.V.: Casp9 target classification. Proteins: Structure, Function, and Bioinformatics 79, 21–36 (2011)
9. Cui, Y., Chen, R.S., Hung, W.: Protein folding simulation with genetic algorithm and supersecondary structure constraints. Proteins: Structure, Function and Genetics 31, 247–257 (1998)
10. Unger, R., Moult, J.: Genetic algorithms for protein folding simulations. Biochim. Biophys. 231, 75–81 (1993)
11. Zhang, G., Han, K.: Hepatitis c virus contact map prediction based on binary strategy. Comp. Biol. and Chem. 31, 233–238 (2007)
12. Konak, A., Coit, D.W., Smith, A.E.: Multi-objective optimization using genetic algorithms: A tutorial. Reliability Engineering and System Safety 91(9), 992–1007 (2006)
13. Judya, M.V., Ravichandrana, K.S., Murugesan, K.: A multi-objective evolutionary algorithm for protein structure prediction with immune operators. Comp. Methods in Biomechanics and Biomedical Engineering 12(4), 407–413 (2009)

14. Calvo, J.C., Ortega, J.: Parallel protein structure prediction by multiobjective optimization. Parallel, Distributed and Network-based Processing 12(4), 407–413 (2009)
15. Shi, S., Suganthan, N.: Parallel protein structure prediction by multiobjective optimization. KanGAL Report 7, 1–7 (2004)
16. Cutello, V., Narzisi, G., Nicosia, G.: A multi-objective evolutionary approach to the protein structure prediction problem. J. R. Soc. Interface 3, 139–151 (2006)
17. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) PPSN VI 2000. LNCS, vol. 1917, pp. 849–858. Springer, Heidelberg (2000)
18. Kyte, J., Doolittle, R.F.: A simple method for displaying the hydropathic character of a protein. J. J. Mol. Bio. 157, 105–132 (1982)
19. Grantham, R.: Amino acid difference formula to help explain protein evolution. J. Mol. Bio. 185, 862–864 (1974)
20. Klein, P., Kanehisa, M., DeLisi, C.: Prediction of protein function from sequence properties: Discriminant analysis of a data base. Bioch. Bioph. 787, 221–226 (1984)
21. Dawson, D.M.: The Biochemical Genetics of Man. Brock, D.J.H., Mayo, O., eds. (1972)
22. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: An update. SIGKDD Explorations 11 (2009)
23. Fernandez, M.A., Paredes, A.B., Ortiz, L.R., Rosas, J.L.: Sistema predictor de estructuras de proteínas utilizando dinámica molecular (modypp). Revista Internacional de Sistemas Computacionales y Electrónicos, 6–16 (2009)

In Silico Infection of the Human Genome

W.B. Langdon and M.J. Arno

Dept. of Computer Science, University College,
London, Gower Street, WC1E 6BT, UK

Abstract. The human genetic sequence database contains DNA sequences very like those of mycoplasma bacteria. It appears such bacteria infect not only molecular Biology laboratories but their genes were picked up from contaminated samples and inserted into GenBank as if they were homo sapiens. At least one mouldy EST (Expressed Sequence Tag) has transferred from online public databases on the Internet to commercial tools (Affymetrix HG-U133 plus 2.0 microarrays). We report a second example (DA466599) and suggest there is a need to clean up genomic databases but fear current tools will be inadequate to catch genes which have jumped the silicon barrier.

Keywords: Bioinformatics, data cleansing, bit rot, *in silico* biology, meme, Blast, phishing, jumping information genes, *in silico* evolution.

1 Introduction

Figure 1 shows how our understanding of genetics has changed over the last 150 years. In each frame the small blue double helix is used to illustrate the movement of genes. In 1865 Mendel showed that genes are discrete units. Originally it was thought that genes were inherited only from parents, however it is now known that genes can be transferred horizontally. Firstly in 1930 McClintock showed

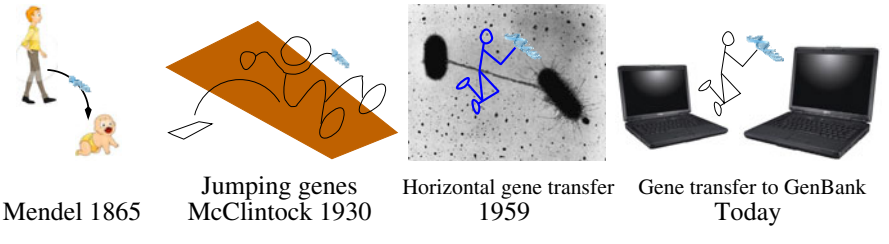


Fig. 1. Tetratych showing: 1865 Mendel's [1] discovery of the essential digital nature of inheritance; 1930 Barbara McClintock's [2] discovery of transposons in Maize whereby genes move not only from parent to child but also along chromosomes; 1959 Micrograph of genetic transfer along a pilus linking two bacteria (Akiba and Ochia discovered the first interspecies gene transfer [3]); mycoplasma bacteria genes are transferred between computers, including into the reference human genome DNA sequence held by GenBank [4]

M. Giacobini, L. Vanneschi, and W.S. Bush (Eds.): EvoBIO 2012, LNCS 7246, pp. 245–249, 2012.

genes could move to new positions in chromosomes of the same species. Later it was found that jumping genes could be transferred between species [3]. Indeed today lateral gene transfer mediated by viral agents is thought to be common [5].

What so far has been little recognised is that jumping genes have escaped biology and now roam our computer systems.

2 The Human Genome

Ensuring databases are both up to date and contain only correct data is a huge software engineering problem. Even as the human genome was first published the associated problems of data cleansing Bioinformatics sequence data were being discussed [6,7] but it appears only technical problems where considered.

We discovered that GenBank, the definitive publicly accessible database holding the human DNA sequence, has been corrupted in a surprising way. It contains the DNA sequence of a bacteria [4].

Section 3 describes how we recently discovered a second sequence which is probably not human in the human genome [8]. Here we extend RN/11/14 [8].

It appears that not only has the human DNA sequence been "completely sequenced" [6] but in the process other living organisms commonly found in molecular biology laboratories have infected not just the physical samples but also the virtual *in silico* Bioinformatics environment. By unwittingly using a technique reminiscent of computer hacking, a bacteria gene has succeeded in not just moving within its own genome [2] nor only jumping horizontally and crossing the species barrier [3] but has crossed the silicon barrier between life and data and succeeded in reproducing itself across very diverse information based media. Given the highly interconnected nature of genomic research, technology and medicine and the low priority so far attached to the problem, it is unlikely current data warehouse cleansing techniques will be able to eradicate this and potentially other silicon jumping genes.

3 Computational *In Silico* Experiment

Using Blast [9] at the European Bioinformatics Institute with their default settings, we searched for the anomalous HG-U133 +2 gene sequence (GenBank AF241217, probeset 1570561_at, which we reported in [4]). This gave a list of DNA sequences which partially match published DNA sequences. The list is ordered by blastn so that the best matches are at the top. Only the top 50 fuzzy matches are included. As expected the first match is the query sequence itself (EM_HTG:AF241217). Despite [4] having been published more than a year ago, EM_HTG:AF241217 is still described as "Homo sapiens". All the others are mycoplasma, except the 34^{th} in the list, DA466599, which EBI says is human. (EBI gives one reference for DA466599: [10]. DA466599 was uploaded to the DNA

Data Bank of Japan 2 years after the HG-U133 +2 was launched). However we suggest that DA466599 may not be a human DNA sequences but is another example of physical contamination leading to virtual infection of the public data.

We ran a second EBI blastn query (again using the NCBI em_rel database). This time looking for DNA sequences that match DA466599. The results for DA466599 are similar to those for AF241217 and so support the view that DNA sequence DA466599 is not human but instead is also a contamination. Again the best 50 matches were reported. Of course the first one is DA466599 itself. All the other matches returned by blastn are for various species of mycoplasma.

4 No *In Silico* Evolution?

Notice what these mycoplasma genes have done. Not only, despite rigorous hygiene standards, do they routinely spread themselves through microbiological laboratories [11] but now at least two have got themselves copied into GenBank and one has spread from there into an Affymetrix GeneChip design. How is this different from any other case where a gene has been sequenced? Fundamentally it is the same. But notice, even though we can post hoc guess a mechanism, it is as if the gene had acted to spread itself. In Biology, gene DNA sequences are acted upon by many mechanisms that copy them but we still adopt the short hand of saying the gene has spread itself [12].

It is difficult to know the number of copies of the human genome. However if we ignore the small number of mirror sites and assume everyone downloads sequences from GenBank directly. This means the GenBank's Internet bandwidth limits the number of copies. Since each copy takes 2 hours, the maximum downloads per year is 4383. Although new versions of GenBank are released "every two months" GenBank is fairly stable and people may not need to be fully up to date, therefore we suggest each copy lasts about a year. This gives an estimate of the global population of the human genome DNA sequence of less than 4 000.

In biology none of the DNA copying mechanisms is perfect. This ensures inherited material is subject to variation. In our computer systems it is often assumed copies are perfect. Indeed we have seen no evidence, yet, of DNA sequences being corrupted once they have been captured by our databases. However changes are possible. Rosenthal [13] says "1.2 10^{-9} of the data written to CERN's storage was permanently corrupted within six months". Also error rates on transferring data across the Internet are never better than 10^{-12} [14] and wireless connections to portable devices are very much worse. Even in the best cases, operator error is always a hazard [14]. In other words, error rates in the best computer systems are much less than typical mutation rates but accidental changes are possible, particularly with portable laptop computers.

After reproduction and variation, the third requirement of evolution is selection. Although one might see human imposed differential selection on corrupted gene sequences, the most likely selection pressure would be simply aimed at removal of errors. Complete extermination would not lead to evolution. Partial erasure might serve. However given the small population size and hence low

copy rate, very low mutation rates and absence of suitable fitness selection, the conditions for the evolution of these *in silico* genes are currently poor.

5 Discussion

It is well known that mycoplasma contamination is rife [11]. Many labs are routinely periodically sterilised to counter it. Miller *et al.* [11] said mycoplasma contamination has "potentially major consequences for the diagnosis and characterization of diseases using expression array technology." Even so, using RNAnet `http://bioinformatics.essex.ac.uk/users/wlangdon/rnanet/`, we previously estimated about 1% of published data in the Gene Expression Omnibus (GEO) database at NCBI (`www.ncbi.nlm.nih.gov/geo`) are contaminated [4].

One potential fortuitous side effect of the *in silico* spread of mycoplasma contamination is that the Affymetrix HG-U133 +2 1570561_at probeset might be used to indicate physical sample contamination. Thus probeset 1570561_at could be treated as a free additional quality control signal. If 1570561_at says there is significant expression of mycoplasma genes, then the sample is probably contaminated and the other gene expression levels given by the microarray are suspect.

Having found two suspect DNA sequences it seems likely the published "human genome" sequence contains more. Indeed contamination of all organism sequences seems possible [15]. With the explosive growth of genomic sequence data available via the Internet, including data from the 1000 genome project [16], it seems time to look again at genomic database quality.

References

1. Mendel, G.: Experiments in plant hybridization. Verhandlungen des naturforschenden Vereines in Brno (IV), 3–47 (1865); Translated by William Bateson
2. McClintock, B.: A cytological and genetical study of triploid maize. Genetics 14(2), 180–222 (1929)
3. Akiba, T., et al.: On the mechanism of the development of multiple-drug-resistant clones of Shigella. Japanese Journal of Microbiology 4, 219–227 (1960)
4. Aldecoa-Otalora, E., Langdon, W.B., Cunningham, P., Arno, M.J.: Unexpected presence of mycoplasma probes on human microarrays. BioTechniques 47(6), 1013–1016
5. McDaniel, L.D., et al.: High frequency of horizontal gene transfer in the oceans. Science 330(6000), 50 (2010)
6. Felsenfeld, A., et al.: Assessing the quality of the DNA sequence from the human genome project. Genome Research 9, 1–4 (1999)
7. Apweiler, R., et al.: Technical comment to "Database verification studies of SWISS-PROT and GenBank" by Karp et al. Bioinformatics 17(6), 533–534 (2001)
8. Langdon, W.B., Arno, M.J.: More mouldy data: Another mycoplasma gene jumps the silicon barrier into the human genome. ArXiv e-prints (June 14, 2011)
9. Altschul, S.F., et al.: Gapped BLAST and PSI-BLAST a new generation of protein database search programs. Nucleic Acids Res. 25(17), 3389–3402 (1997)

10. Kimura, K., et al.: Diversification of transcriptional modulation: Large-scale identification and characterization of putative alternative promoters of human genes. Genome Research 16, 55–65 (2006)
11. Miller, C.J., et al.: Mycoplasma infection significantly alters microarray gene expression profiles. BioTechniques 35(4), 812–814 (2003)
12. Dawkins, R.: The Selfish Gene. Oxford University Press, Oxford (1976)
13. Rosenthal, D.: Keeping bits safe: How hard can it be? Commun. ACM 53, 47–55
14. Handley, M.: Why the internet only just works. BT Technology Journal 24(3)
15. Longo, M.S., et al.: Abundant human DNA contamination identified in non-primate genome databases. PLoS ONE 6(2), e16410 (2011)
16. Durbin, R.M., et al.: A map of human genome variation from population-scale sequencing. Nature 467(7319), 1061–1073 (2010)

Improving Phylogenetic Tree Interpretability by Means of Evolutionary Algorithms

Francesco Cerutti[1,2], Luigi Bertolotti[1,2], Tony L. Goldberg[3], and Mario Giacobini[1,2]

[1] Department of Animal Production Epidemiology and Ecology, Faculty of Veterinary Medicine, University of Torino, Italy
[2] Molecular Biotechnology Center, University of Torino, Italy
[3] Department of Pathobiological Sciences, School of Veterinary Medicine, University of Wisconsin-Madison, USA
{francesco.cerutti,luigi.bertolotti,mario.giacobini}@unito.it, tgoldberg@vetmed.wisc.edu

Abstract. A recent research article, entitled *Taxon ordering in phylogenetic trees: a workbench test* presented the application of an evolutionary algorithm to order taxa in a phylogenetic tree, according to a given distance matrix. In previous articles, the authors introduced the first approaches to study the influence of algorithm parameters on the efficacy of finding the tree with the shortest distance among taxa, based on genetic distances. In the considered work, the authors tested the algorithm using both genetic and geographic distances, and a combination of the two, on three phylogenetic trees of different viruses. The results were interesting, especially when applying geographic distances, allowing a new reading direction, orthogonal to the classical root-to-taxa one.

Keywords: Evolutionary algorithm, phylogenetic tree, taxon order.

1 Short Background in Phylogenetics

Evolutionary biology often makes use of phylogenetic trees to describe and infer the relationships among living organisms. A phylogenetic tree is a mathematical structure representing the evolutionary history of sequences or individuals. It consists of nodes connected by branches, and the terminal nodes, representing the "leaves" of the tree, are called taxa. Internal nodes represent ancestors, and can be connected to many branches. Evolutionary information is contained in the tree topology: in other words, the relationship between two individuals is described by the pathway linking the two tips, along the branches, through the internal nodes. For this reason the most important feature of a tree is its topology.

There are several ways to draw a phylogenetic tree: it is strictly depending by the analyses' aim, but scientists often depict the tree topology as cladogram or phylogram. Basically, the tree is drawn following a horizontal direction where the evolution is described by the pathway from the root of the tree to its tips.

M. Giacobini, L. Vanneschi, and W.S. Bush (Eds.): EvoBIO 2012, LNCS 7246, pp. 250–253, 2012.

Usually, the root of the tree is placed at the left side of the figure and the tips are at the right side. Considering what we stated above, the vertical order in which taxa are reported is not meaningful, since the reading direction is only from root to tips and viceversa, the branch length being a measure of divergence. In fact, taxa belonging to an unresolved clade (where several taxa are linked to the same internal node) are often reported following the same order than in the original input file. This representation is hazardous because a superficial browse through the tree could lead to incorrectly consider the closeness among taxa.

A first approach to reorder the taxa according to a distance matrix was proposed by Moscato, Cotta and colleagues [1,2]. In these works, the authors both build new phylogenies and improve existing ones generated by Neighbor Joining and hypercleaning methods. They approached the problem as a minimum Hamiltonian path problem, and used memetic algorithms to find the "solution that minimizes the length of a path of distances between species" [2].

2 Taxon Ordering in Phylogenetic Trees: A Workbench Test

A more recent article, entitled *Taxon ordering in phylogenetic trees: a workbench test*, published on BMC Bioinformatics [3], described the validation of an Evolutionary Algorithm (EA) to order taxa in a phylogenetic tree given a distance matrix. The idea behind this approach is the following: each internal node in a tree can be freely rotated without modifying the topology. In order to better represent the tree, one could group taxa with similar features, such as genetic similarity, geographic location or collection date, preserving the original topology. This approach was intended to improve the interpretability of phylogenetic trees including more information, especially in highly unresolved trees, and to assist in reading them correctly.

In a previous work [4], the authors investigated the influence of the different parameters on the dynamics of the proposed alforithm. First, a simple $(1 + 1)$-EA was adopted, applying genetic distances for the fitness evaluation. This was considered as the sum of the *vertical distances'* of the r closes taxa to the considered one, for each taxon on the tree. The study proved that the parameter r, called the radius, drastically influenced the algorithm's performances, and a value of $r = 8$ could be a good choice for the fitness evaluation. Comparing the results of the EA with a random search, the former consistently outperformed the latter. Then, the study was directed to the comprehension of the influence of the population size on the search dynamics.The best performances coupled with the more consistent results were obtained when applying $(1 + 5)$-EAs and $(5 + 5)$-EAs.

After this first test to determine the effectiveness of the algorithm and its parameters, the authors validated the method by applying it to three different phylogenetic trees from literature, using both genetic and geographic distances, and a merge of the two.

When reordering the taxa, the trees obtained considering geographic distances showed interesting interpretations. Fig. 1 summarizes the best trees obtained reordering the taxa according to geographic distances and its combination with the genetic ones. The pattern of the points' distribution along the map, representing the State of sample collection, is the same as the one along the tree. Thus, the algorithm effectively reorder taxa on the tree with respect to the distance matrix. The color distribution on the map and on the tree strongly helps the interpretation of the tree, adding a further interpretation.

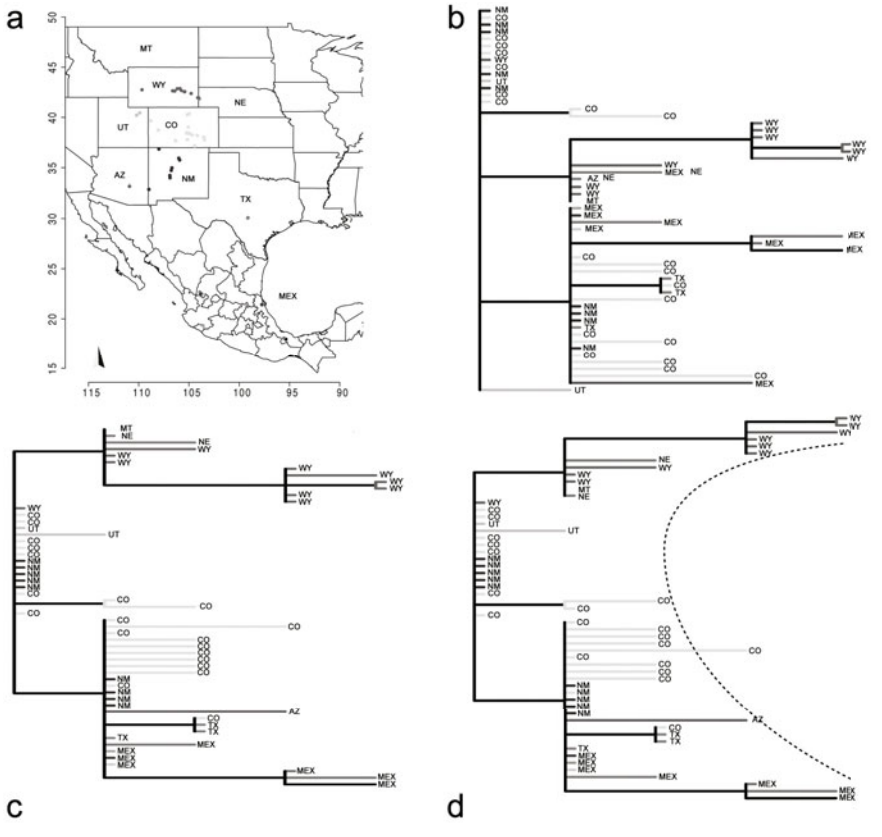


Fig. 1. (a) Map representing the study area of USA and Mexico where VSV samples were collected, and (b) the original tree, as presented by Perez et al.[5]. The best trees obtained using the geographic (c) and combined genetic-geographic (d) distances. The dashed line in D highlights the "C" shape acquired by the clades (the figure is taken from [3]).

When the genetic distances are used, a recurrent reorder occurs, with long branches of the tree pushed to the extremities of the tree. Being the branch

length proportional to the genetic distance, it is correct that the EA reduces the global distance in the tree by moving them to the extremities, since the samples on those branches are the most divergent on the tree.

Other very interesting results in reordering the taxa according to both genetic and geographic distances were obtained by applying the EA to a West Nile virus tree. The samples were collected from a small area in the Cook county, IL, USA, and there was no evident relationship among genetic and geographic distance. Although, genetic variation was larger within sites than between different collection sites. These relationship had a support of the results reported in the considered article. In fact, while with geographic-only and geographic-genetic distances a grouping of samples collected from the same site was recorded, this movement does not appear when applying genetic-only distances.

3 Conclusions

The work presented in the article *Taxon ordering in phylogenetic trees: a workbench test* [3] showed interesting results for helping the interpretation of phylogenetic trees, a new reading direction, orthogonal to the classical root-to-taxa one. The preliminary results of the study were promising, even thou the genetic information is already contained within the tree topology. Adding more information to the tree by using the geographic distance could provide a strong support to the interpretation of phylogenetic trees. The recent development of tools for phylogeography underlines the increasing interest towards the understanding of the relationship among genetic diversity and spatial distribution. The algorithm presented in the article and here discussed does not pretend to be one of them, but is a simpler method to merge the information from genetic and spatial data.

References

1. Moscato, P., Buriol, L., Cotta, C.: On the analysis of data derived from mitochondrial DNA distance matrices: Kolmogorov and a traveling salesman give their opinion. In: Advances in Nature Inspired Computation: The PPSN VII Workshops 2002, pp. 37–38 (2002)
2. Cotta, C., Moscato, P.: A memetic-aided approach to hierarchical clustering from distance matrices: application to gene expression clustering and phylogeny. Biosystems 72, 75–97 (2003)
3. Cerutti, F., Bertolotti, L., Goldberg, T.L., Giacobini, M.: Taxon ordering in phylogenetic trees: a workbench test. BMC Bioinformatics 12, 58 (2011)
4. Cerutti, F., Bertolotti, L., Goldberg, T.L., Giacobini, M.: Taxon ordering in phylogenetic trees by means of evolutionary algorithms. BioData Mining 4, 20 (2010)
5. Perez, A.M., Pauszek, S.J., Jimenez, D., Kelley, W.N., Whedbee, Z., Rodriguez, L.L.: Spatial and phylogenetic analysis of vesicular stomatitis virus over- wintering in the United States. Preventive Veterinary Medicine 93(4), 258–264 (2010)

Author Index