Lecture Notes in Computer Science 7833

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

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

Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics

11th European Conference, EvoBIO 2013
Vienna, Austria, April 3-5, 2013
Proceedings

Springer

Volume Editors

Leonardo Vanneschi
ISEGI, Universidade Nova de Lisboa
1070-312 Lisboa, Portugal and
D.I.S.Co., University of Milano-Bicocca
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

Mario Giacobini
University of Torino
Department of Veterinary Sciences
and Molecular Biotechnology Center
via Leonardo da Vinci 44, 10095 Grugliasco (TO), Italy
E-mail: mario.giacobini@unito.it

Front cover EvoStar 2013 logo by Kevin Sim, Edinburgh Napier University

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-37188-2 e-ISBN 978-3-642-37189-9
DOI 10.1007/978-3-642-37189-9
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2013933105

CR Subject Classification (1998): J.3, H.2.8, I.2.6, F.2, F.1, G.2

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 routinely generate new information on such a massive scale that computer science techniques are now becoming indispensable for translating the resulting data into new biological knowledge. As a consequence, biologists now face the challenges of algorithmic complexity and tractability, as well as combinatorial explosion when conducting even basic analyses. The goal of the 11^{th} European Conference on Evolutionary Computation, Machine Learning, and Data Mining in Computational Biology (EvoBIO 2013) was to bring together experts across multiple fields to discuss new and novel methods for tackling complex biological problems, and often these experts draw inspiration from biological systems in order to produce solutions to biological problems.

The 11^{th} EvoBIO conference took place at the Vienna University of Technology, Austria, during April 3–5, 2013. It was held jointly with the 16^{th} European Conference on Genetic Programming (EuroGP 2013), the 13^{th} European Conference on Evolutionary Computation in Combinatorial Optimization (EvoCOP 2013), the 11^{th} European Conference on Evolutionary and Biologically Inspired Music, Sound, Art, and Design (EvoMUSART 2013), and the European Conference on the Applications of Evolutionary Computation (EvoApplications 2013). Collectively, these events are organized under the name EVO* 2013 (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 2013 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.

First and foremost, we thank all the authors who spent time and effort to generate the excellent contributions to this body of work. We thank the members of the Program Committee for their hard work and expert evaluation in reviewing the submitted papers. We also thank many members of the EVO* community, whose tireless work ensures a smooth and successful conference event; Jennifer Willies from the Edinburgh Napier University, UK, for her unwavering dedication as event coordinator, A. Şima Uyar from the Istanbul Technical University, Turkey, for arranging exceptional publicity for the EVO* events, and Kevin Sim from the Edinburgh Napier University, UK, for his excellent work as webmaster. From the Algorithms and Data Structures Group of the Institute of Computer Graphics and Algorithms at the Vienna University of Technology,

we thank Doris Dicklberger and Günther Raidl for their excellent planning as local organizers and especially Bin Hu for his outstanding work as local Organizing Committee Chair. We extend our gratitude to the Vienna University of Technology for hosting our event, and to Marc Schoenauer from INRIA, France, and to the MyReview team@csregistry.org for providing the publication management system and technical support. Last but not least, we thank our sponsoring organizations: the Algorithms and Data Structures Group at the Institute of Computer Graphics and Algorithms of the Vienna University of Technology, the Austrian Institute of Technology, and the Institute for Informatics and Digital Innovation at Edinburgh Napier University, UK.

We hope you enjoy the excellent research articles included in this volume, and we invite you to contribute to EvoBIO 2014.

April 2013

Leonardo Vanneschi
William S. Bush
Mario Giacobini

Organization

EvoBIO 2013, together with EuroGP 2013, EvoCOP 2013, EvoAPPLICATIONS 2013, and EvoMUSART 2013 was part of EVO* 2013, Europe's premier co-located events in the field of evolutionary computing.

Program Chairs

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

Local Chair

Bin Hu	Vienna University of Technology, Austria

Local Organizers

Doris Dicklberger	Vienna University of Technology, Austria
Günther Raidl	Vienna University of Technology, Austria

Publicity Chair

A. Şima Uyar	Istanbul Technical University, Turkey
Kevin Sim	Edinburgh Napier University, UK

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

Wolfgang Banzhaf	Memorial University of Newfoundland, Canada
Luigi Bertolotti	University of Torino, Italy
Jacek Blazewicz	Poznan University of Technology, Poland
Erik Boczko,	Vanderbilt University, USA
José Caldas	INESC-ID Lisboa, Portugal
Dominique Chu	University of Kent, UK
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
Rosalba Giugno	University of Catania, Italy
Casey Greene	Dartmouth College, Hanover, USA
Jin-Kao Hao	University of Angers, France
Tom Heskes	Radboud University, Nijmegen, The Netherlands
Ting Hu	Dartmouth College, Hanover, USA
Mehmet Koyuturk	Case Western Reserve University, USA
Michael Lones	University of York, UK
Penousal Machado	University of Coimbra, Portugal
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, UK
Alison Motsinger-Reif	University of North Carolina Raleigh, USA
Vincent Moulton	University of East Anglia, UK
Giuseppe Nicosia	University of Catania, Italy
Carlotta Orsenigo	Politecnico di Milano, Italy
Paolo Provero	University of Torino, Italy
Michael Raymer	Wright State University, USA
Marylyn Ritchie	The Pennsylvania State University, USA
Raul Giraldez Rojo	Pablo de Olivade University, Spain
Simona Rombo	ICAR-CNR, Italy
Marc Schoenauer	LRI- Université Paris-Sud, France
Ugur Sezerman	Sabanci University, Turkey
Sara Silva	INESC-ID Lisboa, Portugal
Marc Smith	Vassar College, USA
El-Ghazali Talbi	Univ. des Sciences et Technologies de Lille, France
Stephen Turner	University of Virginia, USA

Ryan Urbanowicz	Dartmouth College, Hanover, 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
Jia Zhenyu	University of California, USA

Sponsoring Institutions

- The Institute of Computer Graphics and Algorithms , Vienna University of Technology, Austria
- The Austrian Institute of Technology
- The Institute for Informatics and Digital Innovationg, Edinburgh Napier University, UK

Table of Contents

Oral Contributions

Poster Contributions

Multiple Threshold Spatially Uniform ReliefF for the Genetic Analysis of Complex Human Diseases

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. Detecting genetic interactions without running an exhaustive search is a difficult problem. We present a new heuristic, multiSURF*, which can detect these interactions with high accuracy and in time linear in the number of genes. Our algorithm is an improvement over the SURF* algorithm, which detects genetic signals by comparing individuals close to, and far from, one another and noticing whether differences correlate with different disease statuses. Our improvement consistently outperforms SURF* while providing a large runtime decrease by examining only individuals very near and very far from one another. Additionally we perform an analysis on real data and show that our method provides new information. We conclude that multiSURF* is a better alternative to SURF* in both power and runtime.

Keywords: Relief, Genetics, Interaction, Epistasis, Heuristic, Weighting, Real Data.

1 Introduction

A perpetual problem in modern genomics is dealing with massive quantities of data. Many geneticists are trying to develop phenotype prediction algorithms for use in clinical practice. However, few such tests have been successfully found. The inherent problem is that most Genome Wide Association Studies (GWAS) so far have focused on linear genotype-to-phenotype association. Whereas there are some phenotypes with strong associations to single gene main effects, the vast majority of possible associations are between multiple genes and one phenotype. Given a phenotype there are only n possible main effects, one for each gene; there are, however, $2^n - n$ possible higher-order signals that are comprised of a set of genes non-linearly correlated with the phenotype. There is an increasing amount of evidence that most genetic associations are in fact high-order and non-additive. [8,1,7].

Given that we would like to find these associations, it then seems that we must venture into combinatorial space. One common way to search for interacting genes is to use the exhaustive search known as Multifactor Dimensionality

L. Vanneschi, W.S. Bush, and M. Giacobini (Eds.): EvoBIO 2013, LNCS 7833, pp. 1–10, 2013.

Reduction (MDR) [6]. Multifactor Dimensionality Reduction works by exhaustively analyzing all possible models of up to n attributes – commonly Single Nucleotide Polymorphisms (SNPs).[1] Searching for two-way or pair-wise interactions takes at least $O(a^2n)$ time in a dataset with a attribute SNPs and n individuals. This is completely unfeasible on a realistically sized genetic dataset with $a >> 10^5$, given modern computing technology and pace of development. Therefore any polynomial-time heuristic that can help find gene interactions is very valuable. One standard approach for finding interacting genes is known as 'two-pass' and involves filtering the dataset by eliminating genes that are considered unimportant by a heuristic. Once the filtering has occurred, the second pass does an exhaustive all-subsets analysis on the much smaller and more manageable new dataset [10]. One of the most successful filters is the SURF* algorithm, developed by Greene et al. in 2010 [2]. We introduce a new improvement, Multiple Threshold SURF (multiSURF*) that further increases the power and vastly decreases the runtime. This will be very valuable for genetics studies, as our new algorithm can process much more data and find more interesting genes in the same time as would be previous necessary for SURF*. This best of both worlds improvement is highly desirable in computer science; SURF* improved upon SURF, but takes much more time. Often amplification algorithms can be used to provide better results given more time, but we provide better results in less time.

1.1 Related Work

In 1994, Kononenko developed ReliefF [5]. Based on previous work by Kira et al. [4], ReliefF proved a novel way of approaching signal detection in binary outcome datasets. The principle underlying the ReliefF algorithm is simple, pick individuals similar to one another and then check if the disease status changes. If it does, reward the attributes that are different and if not reward the ones that are the same. Greene et al. improved upon ReliefF with Spatially Uniform ReliefF (SURF) in 2009 and then again with SURF* in 2010 [3,2]. SURF differs from ReliefF by using all neighbors below a threshold and SURF* uses all neighbors, but splits them into near and far. The attraction of SURF* is that it detects higher-order genetic interactions with no main effects much more often than SURF, even in large datasets. SURF* also takes only $O(an^2)$ time. These two properties combined enable the filtering of huge genetic datasets into ones small enough for an exhaustive search. [2]. We introduce an algorithm that can outperform even SURF*'s power, and also slashes SURF*'s running time. This enhances SURF*'s two attractive properties and makes our alternative better suited for use as a filter in two-pass genetic studies.

[1] SNPs are single letter differences in DNA, each version – or mutation – being called an allele. Since there are two alleles, the original, and the polymorphism with one nucleotide different, there are 4 possible combinations in diploid DNA. Two of these combinations are identical as order is unimportant and we are left with three possibilities for any SNP measurement.

2 Methods

2.1 Simulated Data

Our data were identical to those used to benchmark SURF* and SURF; hence we were able to provide a fair comparison [2].

We used simulated data with each dataset containing one two-way epistatic model and 998 random noise SNPs [11]. Our datasets were balanced and each individual had 1000 SNP measurements with only two as part of a non-linear association with the binary disease status. There was no association between either model SNP and the disease status, but instead a non-linear association involving both SNPs together. Each SNP had a minor and major allele with occurrence frequencies of 0.6 and 0.4, respectively. To generate a dataset we need a penetrance function to determine genotype phenotype relationship. We used 30 penetrance functions that met the requirements of having no main effects and the proper minor allele frequencies[2]. Each dataset also had a heritability. This is the probability that a genetic effect will carry through to a phenotype. A heritability of 1.0 means a perfect association between genotype and phenotype, whereas 0.0 is no association. Our datasets were generated with heritabilities of 0.025, 0.05, 0.1, 0.2, 0.3, and 0.4; we used 5 penetrance functions for each. For each of these 30 models we used 100 random variants of the dataset generated using the same penetrance function and heritability. Finally, datasets with duplicate genetic models were generated with 800, 1600, and 3200 individuals.

2.2 Real Data

This study population consists of nationally available genetic data from 2,286 men of European-descent (488 non-aggressive and 687 aggressive cases, 1,111 controls) collected through the Prostate, Lung, Colon, and Ovarian (PLCO) Cancer Screening Trial., a randomized, well-designed, multi-center investigation sponsored and coordinated by the National Cancer Institute (NCI) and their Cancer Genetic Markers of Susceptibility (CGEMS) program. We focused here on prostate cancer aggressiveness as the endpoint. Between 1993 and 2001, the PLCO Trial recruited men ages 55-74 years to evaluate the effect of screening on disease specific mortality, relative to standard care. All participants signed informed consent documents approved by both the NCI and local institutional review boards. Access to clinical and background data collected through examinations and questionnaires was approved for use by the PLCO. Men were included in the current analysis if they had a baseline PSA measurement before October 1, 2003, completed a baseline questionnaire, returned at least one Annual Study Update (ASU), and had available SNP profile data through the CGEMS data portal (`http://cgems.cancer.gov/`). We used a biological filter to reduce the set of genes to just those involved in apoptosis (programmed cell

[2] This and the experimental design section are synopses of our experiment, which was identical to that from both the SURF and SURF* papers. For another description of the experiment and further explanation of the design, please see references [3,2].

death), DNA repair and antioxidation/carcinogen metabolism. These biological processes are hypothesized to play an important role in prostate cancer. A total of 219 SNPs in these genes were studied here.

2.3 multiSURF* Implementation

Notation. We denote the i^{th} individual from the dataset as I_i; $d(I_i, I_j)$ refers to the hamming distance between two individuals I_i and I_j. The hamming distance is calculated by counting the number of unequal SNP measurements between the two individuals. For instance, an individual with measurements 01001 will have a hamming distance of 2 from an individual with 01010. $s[i]$ refers to the disease status – case or control – of individual I_i. We denote the number of individuals with n and the number of SNPs – or attributes – with a. We denote the standard deviation of a set/vector X as σ_X.

Algorithm. The SURF* algorithm functions by making comparisons between individuals. We first define a weight for each SNP which is initialized to 0. The type of comparison depends on whether an individual is near or far from another. We start by computing the mean of all distances, which we call the threshold, T. Then for each possible non-ordered pairing of individuals I_i, I_j we check if $d(I_i, I_j) < T$. If so we treat individual I_j as near to I_i, and in the opposite case, $d(I_i, I_j) > T$, we treat it as far from I_i. If the two individuals are near to one another we check all SNPs that have different values across both individuals. For each of these we increment the weight if the status is different and decrement if the status is the same. If the two individuals are far from one another we check all SNPs that have the same value across both individuals. For each of these we increment the weight if the status is different and decrement if the status is the same.

The authors note that the original SURF* paper described the algorithm as examining SNPs that were different across far individuals. We implement the algorithm by examining SNPs that are the same and reversing the signs on increment and decrement. This does not affect the monotonicity of the final scores of the SNPs, and is slightly faster.

We change the SURF* algorithm in two distinct ways. First of all we give each individual its own threshold rather than having one global threshold. Second we more strictly define near and far. For more reference to the original SURF* algorithm please see the corresponding paper [2].

Multiple Thresholds. In the SURF* algorithm one global distance threshold T is initially computed. Any individual I_j closer than T to individual I_i is then classified as near and any individual I_j further than T from individual I_i is classified as far. Instead of this global threshold, T, we compute one threshold, T_i, for each individual I_i. In SURF*, T is computed as the mean distance between all $(n^2 - n)/2$ individual pairings I_i, I_j. In multiSURF* we compute each T_i to be the mean of all distances $d(I_i, I_j)$ between the fixed individual I_i and all other individuals I_j. Formally:

$$T_i = \frac{\sum_j d(I_i, I_j)}{n - 1}.$$

Near and Far. In the SURF* algorithm each individual I_j is treated as being either near to, or far from, another individual I_i. This is determined by whether $d(I_i, I_j) < T$. This means that a comparison is made between every pair of individuals. In multiSURF* we have stricter definitions of near and far and therefore make fewer comparisons. While computing T_i we compute the distances between I_i and every other individual and save this as a vector, V. We take the standard deviation of these distances, σ_V and set a new dead-band variable D_i equal to $\sigma_V/2$. We then say that an individual I_j is near to I_i if and only if $d(I_i, I_j) < T_i - D_i$. Similarly, an individual is far if and only if $d(I_i, I_j) > T_i + D_i$. If we assume a normal distribution of distances this means that $\Phi(\sigma/2) \approx 31\%$ of individuals I_j will be classified as near and likewise for far. This reduces the runtime by a large factor, as multiSURF* makes approximately 62% of the comparisons that SURF* does.

Pseudo-code. A pseudo-code description of the algorithm can be found in Algorithm 1. We implemented the algorithm in Java. For full source code, please contact the authors at the given email.

Running Time. SURF* first computes all distances, this takes $c_1 n^2 a$ time for some system-dependent c_1. The next round of computation to update the weights runs over all pairs of instances and takes $c_2 n^2 a$ time for another system-dependent constant. SURF therefore takes a total of $(c_1 + c_2)n^2 a$ time. multiSURF* computes all distances, so again we take $c_1 n^2 a$ time. In addition we compute a D_i for each individual I_i. The required variance computation is linear and takes $c_3 n^2$ in total. Next we perform an instance comparison if and only if the distance between the two satisfies our dead-band requirement. We assume a normal distribution of distances between a given instance I_i and all others. To find the number of instances that will be near or far is the same as computing a two-tailed p-value. Using the cumulative distribution function we compute

$$2(1 - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{1/2} e^{\frac{-x^2}{2}} \, \mathrm{d}x) \approx 0.62$$

Therefore we will perform about $0.62n^2$ comparisons, each taking the same time $c_2 a$. This yields a total runtime of

$$c_1 n^2 a + c_3 n^2 + 0.62 c_2 n^2 a =$$

$$(c_1 a + 0.62 c_2 a + c_3) n^2 =$$

For large values of a we can obtain the very good approximation

$$(c_1 a + 0.62 c_2 a + c_3) n^2 < (c_1 a + 0.62 c_2 (a + c_4)) n^2 \approx$$

$$(c_1 a + 0.62 c_2 a) n^2$$

Compared to SURF*'s

$$(c_1 a + c_2 a) n^2$$

```
Data: Input dataset data
Result: Array of weights w
set d[i][j] = d(I_i, I_j) for all i and all j;
for each i do
    compute all distances between I_i and every other indiviudal;
    set T_i to be the mean of all these distances;
    set σ to be the standard deviation of the distances;
    set D_i to be σ/2;
end
for each i do
    for each j do
        if d(I_i, I_j) < T_I − D_i then
            for each a do
                if I_i[a]! = I_j[a] then
                    if s[i] == s[j] then
                        w[a] − −;
                    else
                        w[a] + +;
                    end
                end
            end
        end
        if d(I_i, I_j) > T_I + D_i then
            for each a do
                if I_i[a] == I_j[a] then
                    if s[i] == s[j] then
                        w[a] − −;
                    else
                        w[a] + +;
                    end
                end
            end
        end
    end
    return w;
end
```

Algorithm 1. multiSURF*

Therefore multiSURF* will outperform SURF* by some system dependent constant for large values of a. To test how the two performed on a modern computer we ran multiSURF* on a sampling of the previously defined datasets. The machine was an Intel Core 2 Duo P8600 2.4GHz with 4GB of RAM. We arbitrarily selected 10 datasets with 1600 instances and a heritability of 0.05 and ran 10 times on each dataset for a total of 100 runs. The average runtime and standard deviation for each method is detailed in Table 1.

2.4 Experimental Design

To measure success we examined the rankings returned by multiSURF* and SURF* for each dataset. For each ranking we chose the minimum score of the two model SNPs and found the percentage of SNPs ranked higher. We then counted the number of times that the model SNPs were ranked in the top $x\%$ over the $\frac{30 \text{ genetic models}}{6 \text{ heritabilities}} \times 100 \text{ variants} = 500$ datasets for each heritability and number of individuals. These results are displayed in Figure 1.

To test if the observed differences were significant we counted the number of times that multiSURF* ranked the model SNPs higher than 95% of all other SNPs over all 9000 datasets. Doing the same for SURF* gave us a contingency table. We used a Fisher's exact test on these counts to obtain a probability that the observed difference was due to chance; the p-value can be found in the results section along with a discussion of the graphical results [9].

2.5 Real Data Analysis

To further validate multiSURF* we performed a comparative analysis on the dataset described in Section 2.2.

We ran several Relief-family methods on the dataset: ReliefF with 350 neighbors, SURF, SURF*, and multiSURF*. We then ranked SNPs by the weight assigned to them. We wished to determine which genes each algorithm treats as significantly more important than others, so we performed a Fisher's exact test. We found the corresponding gene for each SNP and then counted the number of times that gene was represented by a SNP ranked in the top 10%. These counts gave us a p-value for each SNP via the Fisher's exact test. It should be noted that the p-value is not corresponding to any actual model for the dataset, simply whether the genes were being ranked higher by the algorithm by chance. We then selected all genes that had achieved a significant p-value for any Relief method and listed them in a table. The table can be found in Table 2 and shows which genes are being considered 'important' by which algorithm.

3 Results

3.1 Success Rate

The results are presented in Figure 1. The difference between the two methods was found to be highly significant ($p < 10^{-15}$). We found that multiSURF* outperformed SURF* on all sample sizes and heritabilities. The exceptions are the cases in which SURF* already achieves a 100% success rate on all percentiles. The improvements were consistent across all the sample sizes and heritabilities.

3.2 Runtime

The runtimes can be found in Table 1. On average, multiSURF* runs in only 67.8% of the time of SURF*. The code implementation was in Java and not

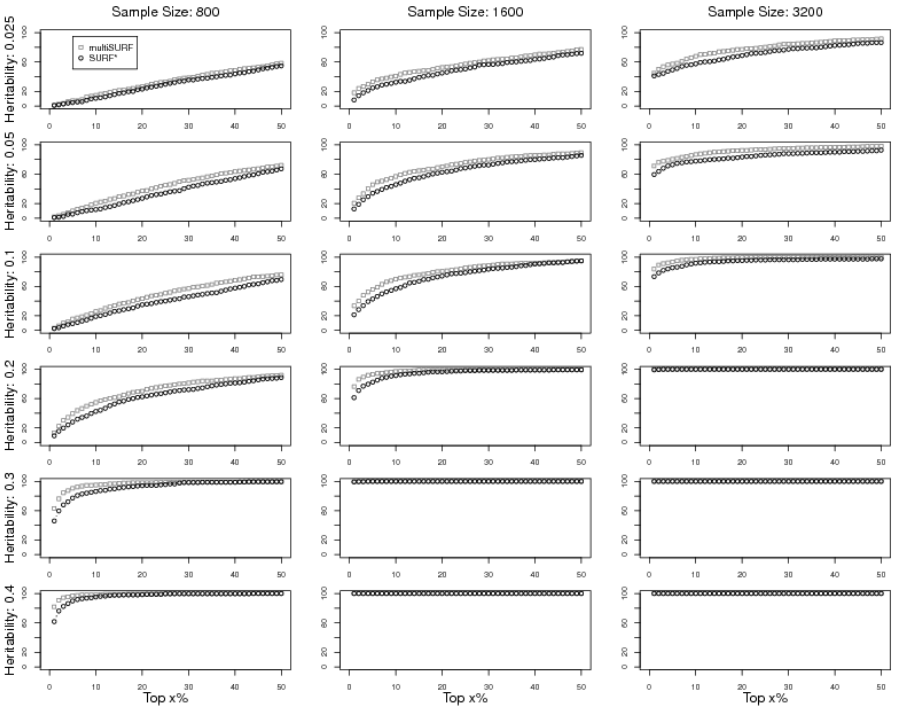


Fig. 1. Success rate of multiSURF* vs. SURF*

Table 1. Comparing the runtimes of multiSURF* and SURF*

	Mean Runtime (s)	Standard Deviation (s)
multiSURF*	8.323	0.369
SURF*	12.276	1.275

optimized, we only wanted to find a comparison between the two methods, implemented identically. A faster implementation using threading could vastly cut down on execution time for both methods.

3.3 Real Data Results

The analysis results are shown in Table 2. The symbol $*$ indicates statistical significance ($p < 0.05$), $**$ indicates high statistical significance ($p < 0.01$). We note that multiSURF* found the same genes as the other algorithms, with the exception of BCL2L11, which was found only by SURF*. In addition, multiSURF* found a new gene, ATK3. Several of the p-values were identical because they were based on discrete counts and a Fisher's Exact test.

Table 2. The probability that each gene received higher rankings by chance

	ReliefF	SURF	SURF*	multiSURF*
NAT2	0.0000106**	0.0005274**	1	0.0000106**
PRKCQ	0.001464**	0.001464**	0.001024**	0.001464**
RAF1	0.0001462**	0.301	1	0.01504*
ATK3	0.106	0.106	1	0.0000196**
BCL2L11	0.2953	0.2953	0.02725*	1

4 Discussion and Conclusions

The increase in power stems largely from the disregarding of the individuals neither near nor far, which is enabled by the individual thresholds. Since the distribution of zygosity measurements $\{0, 1, 2\}$ is likely not uniform for any given SNP and in our case is highly non-uniform, a given individual I_i may have common or uncommon values. Take for example the case in which 0 is common for all SNPs, the individual with measurements $0, 0, \ldots, 0$ will be much closer, on average, to any other individual. An individual with complementary measurements such as $2, 2, \ldots, 2$ will, on the other hand, be much further from other individuals. Therefore in a large random population there will be individuals for which a global threshold is ill-fitting; instead of splitting other individuals evenly with roughly half being classified near and half far, the majority of individuals will fall into one class or the other. This is effectively a dilution of potentially useful information contained in near and far individuals. If we assume a normal distribution of distances from I_i to all other individuals, then having individual thresholds, T_i, set to the mean of the differences will split the instances evenly into near and far. With this even split we can then apply the dead-band and disregard individuals that are neither very near nor very far. By looking only at the extremes of the distribution we can provide a better result for effectively the same reasons that SURF* works in the first place. A rigorous mathematical proof of this has yet to be presented and is a potential topic of future research. The authors examined several arbitrarily chosen values for the deadband multiplier, namely the α for $\alpha \times \sigma$. $\alpha = 1/2$ worked the best for all tested data, but an argument could be made for α being a parameter of the algorithm. In fact, every dataset may have an optimal α for finding important SNPs in that dataset, but a way to quickly determine the optimal α for a given dataset is not currently known.

Overall multiSURF* is a desirable alternative to SURF*; it takes less time to run and will find an underlying genetic model more frequently. This enables geneticists to filter even larger datasets and search for higher-order models with both greater power and efficiency.

Acknowledgments. The authors would like to thank Jeff Kiralis for general assistance and Nadia Penrod and Jeff Kiralis for reviewing the manuscript. This work was funded by NIH grants AI59694, LM009012, LM010098.

References

1. Cordell, J.H.: Detecting genegene interactions that underlie human diseases. Nature Reviews Genetics 489, 392–404 (2009)
2. Greene, C.S., Himmelstein, D.S., Kiralis, J., Moore, J.H.: The Informative Extremes: Using Both Nearest and Farthest Individuals Can Improve Relief Algorithms in the Domain of Human Genetics. In: Pizzuti, C., Ritchie, M.D., Giacobini, M. (eds.) EvoBIO 2010. LNCS, vol. 6023, pp. 182–193. Springer, Heidelberg (2010)
3. Greene, C.S., Penrod, N.M., Kiralis, J., Moore, J.H.: Spatially uniform relieff (surf) for computationally-efficient filtering of gene-gene interactions. BioData Mining 2 (2009)
4. Kira, K., Rendell, L.A.: A practical approach to feature selection. Machine Learning, 249–256 (1992)
5. Kononenko, I.: Estimating Attributes: Analysis and Extensions of RELIEF. In: Bergadano, F., De Raedt, L. (eds.) ECML 1994. LNCS, vol. 784, pp. 171–182. Springer, Heidelberg (1994)
6. 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 (2007)
7. Moore, J.H., Ritchie, M.D.: The challenges of whole-genome approaches to common diseases. JAMA 291, 1642–1643 (2004)
8. Moore, J.H., Williams, S.M.: Epistasis and its implications for personal genetics. AJHG 85, 309–320 (2009)
9. Sokal, R.R., Rohlf, F.J.: Biometry: the principles and practice of statistics in biological research, 3rd edn.
10. Thomas, D.: Gene-environment-wide association studies. Nat. Rev. Genetics 11, 259–272 (2010)
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)

Time-Point Specific Weighting Improves Coexpression Networks from Time-Course Experiments

Jie Tan, Gavin D. Grant, Michael L. Whitfield, and Casey S. Greene

Department of Genetics, The Geisel School of Medicine at Dartmouth,
2 East Wheelock St. Hanover NH 03755, USA
{Jie.Tan.GR,Casey.S.Greene}@dartmouth.edu

Abstract. Integrative systems biology approaches build, evaluate, and combine data from thousands of diverse experiments. These strategies rely on methods that effectively identify and summarize gene-gene relationships within individual experiments. For gene-expression datasets, the Pearson correlation is often applied to build coexpression networks because it is both easily interpretable and quick to calculate. Here we develop and evaluate weighted Pearson correlation approaches that better summarize gene expression data into coexpression networks for synchronized cell cycle time-course experiments. These methods use experimental measurements of cell cycle synchrony to estimate appropriate weights through either sliding window or linear regression approaches. We show that these weights improve our ability to build coexpression networks capable of identifying phase-specific functional relationships between genes. We evaluate our method on diverse experiments and find that both weighted strategies outperform the traditional method. This weighted correlation approach is implemented in the Sleipnir library, an open source library used for integrative systems biology. Integrative approaches using properly weighted time-course experiments will provide a more detailed understanding of the processes studied in such experiments.

Keywords: Functional Genomics, Time-course Experiment, Coexpression Network, Weighted Pearson Correlation.

1 Introduction

Methods that integrate heterogeneous data into "functional networks" provide a systems level portrait of an organism's molecular process [1–5]. These methods have proven to be a powerful way to summarize and interpret high-throughput genome-scale data in the context of high-specificity experimental results. In these networks, each node is a gene and each edge represents the probability that two genes are involved in the same process or pathway. Approaches leveraging these methods have been able to efficiently and effectively direct definitive molecular experiments [6, 7].

L. Vanneschi, W.S. Bush, and M. Giacobini (Eds.): EvoBIO 2013, LNCS 7833, pp. 11–22, 2013.

Functional integration strategies are extremely powerful because they use billions of individual measurements to build comprehensive gene-gene functional networks. For example, Huttenhower et al. [5] used more than 27 billion data points across more than 650 datasets from more than 15,000 publications to build networks. The IMP webserver [8], recently developed for integrative systems biology, makes predictions from more than 2400 experimental datasets. Thus given the amount of data available, these approaches are no longer limited by the breadth of the data (i.e. how many conditions are measured) but instead by the depth of the data (the amount of information in individual measurements). Because these approaches rely on experiments being summarized to gene-gene networks, methods that improve such within-experiment networks will play a key role in the development of integrative strategies that better predict and inform biological experiments.

By building a coexpression network from time-course experiments with an approach that acknowledges the known relationships between time points (see Figure 1), we are able to more effectively identify genes that participate in shared biological processes (i.e. phases of the cell cycle). We develop a strategy capable of effectively using data from time-course experiments to provide time point weights for network building methods. In these experiments, a biological system is surveyed periodically, often in response to a chemical or environmental stimulus. Such experiments represent a commonly used experimental design (at the time of this writing of the 2466 datasets used by the IMP webserver, 380 of them are time-courses [9]).

Here we perform proof-of-concept evaluation of our method on time-course studies of the human cell cycle. Progression through the cell cycle, which includes DNA replication (occurring during S-phase) and mitosis (occurring during M-phase), is tightly regulated in all cells. Understanding the timing and regulation of this process is critical, as the catastrophic results of aberrant or failed cell cycle progression include cancer or programmed cell death. To this end, a number of techniques have been used to study the expression patterns of genes involved in and/or regulating the cell cycle; these include genome wide expression arrays of synchronous cells [10–13] and real time luciferase assays using cell cycle promoters [14, 15]. Each of these techniques provides a time course covering multiple synchronous cell cycles in a population of cells. We show that experimental data from both methods can be combined to produce coexpression networks that better identify gold standard functional relationships than those produced by standard methods.

We expect that our method will be generally applicable to data from different synchronization methods and different cell types. We also expect that our approach will generalize to different types of time-course experiments such as those studying circadian rhythms, which are cyclic and have also been assayed on a genome-scale [16]. Understanding how circadian rhythms function, and thus how they can be modified, will be critical to identifying how humans adjust (or fail to adjust) to rapid travel across time-zones. Time-course experiments represent

a substantial fraction of publicly available experimental results, and inexpensive genome-scale assays based on next-generation sequencing technologies are expected to reduce the cost of such experiments and thus increase the amount of available data. Therefore by developing methods that more effectively use such data, we can develop higher quality integrative methods that more efficiently identify and prioritize candidate-genes for high-specificity molecular assays of gene-function or gene-disease relationships.

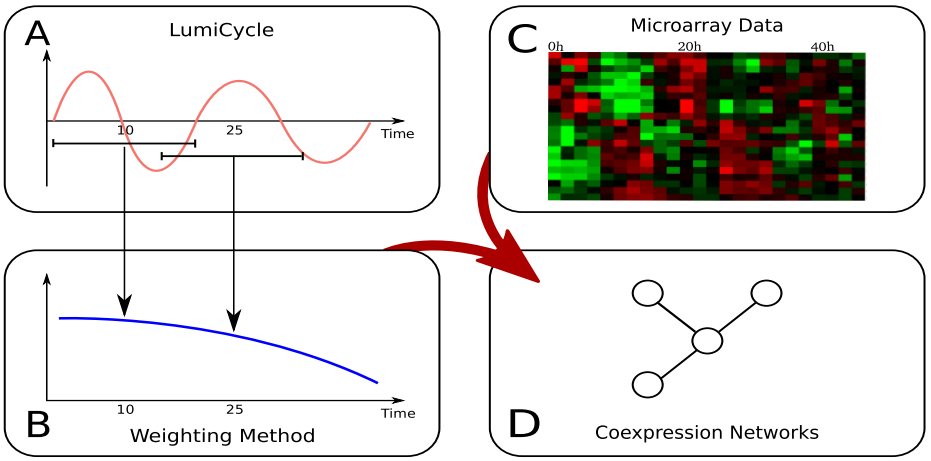


Fig. 1. This diagram outlines how we use experimental assays of cycle synchrony (A) to develop weighting method (B). We use surrounding cycle of each time point in the LumiCycle to calculate the weight at that time point. Then we apply these weights with time-course microarray datasets (C) to build high quality networks of coexpression (D). The strong red and green gene-expression bands at early time points become less synchronized at later time points. The experimentally derived weighting methods (B) allows later measurements where the cell populations are less synchronized to contribute less to coexpression networks.

1.1 Related Work

While computational approaches exist to identify periodically expressed genes in time-course data [17–23], such methods don't provide a global portrait of the relationships between all genes. In contrast, approaches that take a genome-scale view treat each time point as independent, ignoring their connected nature [4, 5, 8]. In this work, we show that combining time-course data with methods for building gene-gene coexpression networks that take into account the temporal relationships between samples provides substantial improvement in the predictive power of such networks. By developing a weighted-Pearson approach, our method provides a drop-in replacement that improves the quality of information that time-course experiments provide to large-scale integrative approaches.

2 Data and Methods

2.1 U2OS LumiCycle Measurements

In order to directly measure the expression of cell cycle regulated genes over the time-course, we used data from luciferase reporter constructs for G1/S and G2/M previously reported by Grant et al. [14]. To measure G1/S phase, a minimal human E2F1 luciferase reporter construct was used [24], and for G2/M a reporter construct with synthetic FOXM1 responsive element that drives luciferase expression was used[14]. Data were collected using LumiCycle (Actimetrics, Wilmette, IL), a real-time luminometer. The LumiCycle provides several advantages over a traditional luminometer. It allows for accurate, simple data collection due to frequent automated measurements. Furthermore, the measurements are non-destructive, meaning the same population of cells is used throughout the experiment. This removes the variability of sample-to-sample comparison and reduces the number of replicates needed.

2.2 U2OS Time-Course Gene-Expression Assays

Data were obtained from genome-scale assays of gene expression in the context of the human cell cycle in U2OS cells. Cells were arrested in a specific cell cycle phase using a chemical treatment, released, and the expression of all genes in the genome were measured across multiple synchronous cell cycles using microarray technology (unpublished data from Grant and Whitfield). Briefly, U2OS osteosarcoma cells were synchronized in G1/S using double thymidine or in M phase using thymidine-nocodazole [14]. RNA was isolated from cells every two hours and hybridized to Agilent Technologies (Santa Clara, CA) Whole Human Genome Oligonucleotide arrays (4x44k) following the manufacturers protocol. Data were analyzed and processed using the same procedures as in Whitfield et al. [13]. Spot identifiers from the microarray were transformed to Entrez GeneIDs using IDconverter [25]. Expression values for spots measuring the same Entrez GeneID were averaged.

2.3 HeLa Time-Course Gene-Expression Assays

To confirm the generality of our findings in a different cell line, we obtained previously published synchronized HeLa time course data from Whitfield et al. [13] for time courses longer than two cell cycles. This resulted in three additional datasets (noted in [13] as Thy-Thy2, Thy-Thy3, and Thy-Noc). In this experiment, HeLa cells were synchronized in either G1/S (TT2, TT3) or M (TN) using double thymidine or thymidine-nocodazole arrest respectively. Total RNA was isolated and arrayed on custom Stanford-printed microarrays. As with the U2OS data, spot identifiers from the microarray were transformed to Entrez GeneIDs using IDconverter [25], and expression values for spots measuring the same Entrez GeneID were averaged.

2.4 Using LumiCycle Measurements to Measure Cycle Synchrony

We developed a measurement of cycle synchrony by using measurements from the LumiCycle. As the populations of cells in the assay become desynchronized, the amplitude is reduced and the standard error of the measurements increases (Figure 2). We developed and evaluated both sliding window and linear regression based approaches.

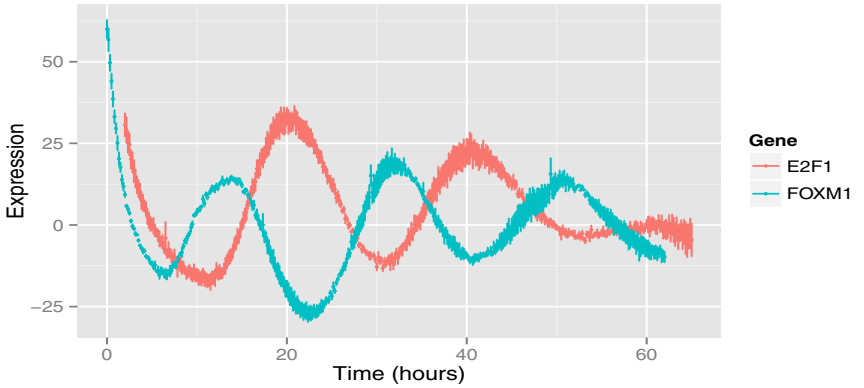


Fig. 2. The expression of reporter constructs as measured by the LumiCycle. Each point represents the mean and the bars represent the standard error from four repeated trials. The E2F1 construct peaks at the G1/S phases and the FOXM1 construct peaks in the G2/M phases.

For the sliding window method, we obtained weights by calculating the ratio of the range of mean expression values for time points within the surrounding cycle (the surrounding 20 hours in U2OS cells) to the mean of the standard errors observed for each time point within that cycle. At the boundary positions, we used the nearest measured portion of the cycle (i.e. the nearest 20 hours to the point of interest). We termed this approach the "Window" method.

For the linear regression method, we calculated, for each LumiCycle measurement, the ratio of the range of mean expression values for time points within the surrounding cycle to the standard error of the LumiCycle measurement in question. This provided a potential weight for each point. We then used least squares linear regression to fit a line to these points. For a vector of microarray assayed time points, we calculated the vector of weights using the linear model. We termed this approach the "Linear" method.

We summarize cell synchrony using the two methods for each reporter (E2F1 or FOXM1). Because LumiCycle measurements of E2F1 begin at 2 hours and measurements for FOXM1 end at 62 hours, measurements from the closest available time point were used beyond these bounds. Then the weight vectors derived from E2F1 and FOXM1 were averaged.

We also showed that our weighting approach generalizes to the HeLa cell line. HeLa cells are biologically quite distinct from U2OS cells, and the HeLa cell cycle lasts only 14 hours [13]. The relationship between U2OS cell cycle and HeLa cell cycle was expected to be linear-scaling. Therefore we performed a linear compression of the time course of LumiCycle of U2OS cells to estimate the corresponding HeLa time course weights. Because each HeLa microarray time point did not directly match a compressed U2OS LumiCycle time point, the average weight of the two closest time points was assigned.

2.5 Construction and Evaluation of Coexpression Networks

The Pearson product-moment correlation coefficient between each pair of genes was calculated based on their expression value. Since the cell cycle synchrony degraded over time, we applied the weighted Pearson correlation using the weight vectors discussed above to achieve more precise coexpression networks. The weighted Pearson correlation was calculated as

$$corr\,(x, y; w) = cov\,(x, y; w)\,/\sqrt{cov\,(x, x; w) \times cov\,(y, y; w)}. \tag{1}$$

$$cov\,(x, y; w) = \frac{\sum_i w_i (x_i - \frac{\sum_i w_i x_i}{\sum_i w_i})(y_i - \frac{\sum_i w_i y_i}{\sum_i w_i})}{\sum_i w_i}. \tag{2}$$

where x and y represent two gene expression vectors, w represents weight vector. We compared networks derived from this weighted approach to control coexpression networks obtained with the standard unweighted Pearson correlation.

Weighted methods to build coexpression networks were evaluated using two gold standards. Both standards assess the ability of the methods to discover relevant phase-specific signal. The importance of a high-quality gold standard for integrative approaches has been well demonstrated [26]. We curated a high specificity gold standard containing 101 genes (`http://discovery.dartmouth.edu/~cgreene/whitfield-curated/cell-cycle-genes.html`) with published cell cycle involvement independent of Whitfield et al. [13] and a very high specificity standard of 15 cell cycle genes (Table 1). The smaller set of gold standard genes are well characterized cell cycle genes that were known to be cell cycle regulated prior to the benchmark publication of Whitfield et al.[13]. Each gold standard was made up of two phase-specific components. Genes in the standard were annotated as involved in either the G1/S or the G2/M cell cycle phases.

To evaluate the ability of our approach to differentiate between such closely related processes, we considered edges between all pairs of genes involved in the same phases (e.g. G1/S) as positives and edges between genes that crossed phases (e.g. an edge between a gene in G1/S and one in G2/M) as negatives. This insured that we evaluated summarization methods on their ability to identify phase-specific edges and not simply to identify pairs of cell cycle genes.

Both unweighted and weighted Pearson correlation results were compared to gold standards using Sleipnir library [27]. We assessed the performance of each weighting strategy (equal, window, and linear) using the area under the curve

Table 1. Genes in the table are either mainly expressed in G1/S phases or G2/M phases and show cycling RNA expression in the cell cycle. They were used to build the highly specific gold standard.

G1/S Phase		G2/M Phase
CCNE1	CCNE2	CCNB1
E2F2	PCNA	BUB1
MCM2	RRM2	CCNB2
RFC4	SLBP	BUB1B
HIST1H4C	HIST1H4B	CDC25B

(AUC) for each gold standard applied to the co-expression network. This evaluated how much phase-specific signal remained in the summarized co-expression networks. Signal remaining after summarization to co-expression networks will be available to further integrative approaches and webservers (e.g. IMP [8] `http://imp.princeton.edu`) that combine such data to answer specific biological questions.

We also calculated the percentage of the phase-specific signal lost under the equal weighting but retained by the weighted approaches. To do this, we assumed that the datasets were capable of perfectly recapitulating all true biology related to the ordering of cell-cycle gene pairs in a coexpression network. With this potential overestimate of recoverable biology, this assumption was conservative and thus may have underestimated the percentage of the signal that we recovered. Therefore results should be interpreted as a lower-bound. We calculated the percentage of the signal retained as

$$\%S_r = (\mathrm{AUC}_W - \mathrm{AUC}_E)/(\mathrm{AUC}_{MAX} - \mathrm{AUC}_E) \times 100. \quad (3)$$

AUC_E was the baseline area under the curve observed with equal weighting, and AUC_W was the AUC observed using the specified weighting approach. Because we made the conservative assumption that these data contained all information necessary to perfectly separate gold standards from these phases, the AUC_{MAX} was set to 1 for our evaluation. If a weighted approach performed worse than the equal weighting approach, this value would be negative indicating a loss in performance by weighting.

3 Results

We developed approaches to convert measurements of cell synchrony (Figure 2) in time-course experiments into weights (Figure 3). These weights are used to build a coexpression network with a weighted Pearson correlation. We show that using these experimentally derived weights better recapitulates the biological system, in this case genes' relationship in the cell cycle, than equal weighting, the current state of the art in the field. We also show that this approach

generalizes beyond the specific cell line used to define the weights. We used measurements from the human U2OS cell line to define the weights, and apply them to previously performed time-course experiments in the human HeLa cell line. The weighted coexpression networks from these HeLa experiments also better recapitulate known biology than their unweighted counterparts. We have implemented the weighted Pearson correlation into the open source Sleipnir library for functional genomics version 3 or above (`http://libsleipnir.bitbucket.org`).

3.1 Time-Course Weights

We used high-density measurements of the expression of reporter constructions from the LumiCycle (results shown in Figure 2) to calculate weights for each individual time point. Weights were determined by either a sliding window or linear regression approach as described in Section 2.4. The calculated weights for each LumiCycle time point are shown in Figure 3. The least-squares linear regression approach identified a best fit line of

$$W(t) = -0.661272585t + 49.7722026115 \tag{4}$$

where t represents the time in hours since the beginning of the experiment.

3.2 Performance of Weighting of U2OS Time-courses

Our methods using weights derived from LumiCycle measurements clearly and consistently outperformed the traditional method using equal weights. Full results are shown in Table 2. This superior performance was observed across all time-course experiments and gold standards. Thy-Thy1, Thy-Thy2, and Thy-Thy3 are the three distinct thymidine-thymidine synchronized time-courses, and Thy-Noc was a thymidine-nocodozole synchronized time-course experiment. While performance was very strong across both gold standards, the proportion of signal retained was most pronounced for the gold standard of phase-specific genes known to show cycling expression. This gold standard represents the situation where our assumption that $\mathrm{AUC}_{MAX} = 1$ is most likely to be true, and thus the situation where our estimate of the proportion of additional signal retained is less likely to be too conservative.

3.3 Generalization to HeLa Time-Course Experiments

Different cell types and cell lines progress through the mitotic cell cycle at different rates. While the U2OS cells progressed through the cell cycle with approximately a 20 hour period, HeLa cells progress through the cycle with a 14 hour period. We thus linearly scaled the time in hours associated with each weight in U2OS to obtain a weight for HeLa cell cycle time-course experiments. We used these weights to build coexpression networks from the HeLa time-course data described in Section 2.3. Performance gains for weighted networks over networks built using the equal weighting approach were more modest than with the U2OS experiments. Still, for every time-course experiment, the approaches using differential weighting were better able to recapitulate known relationships than the equal weight strategy.

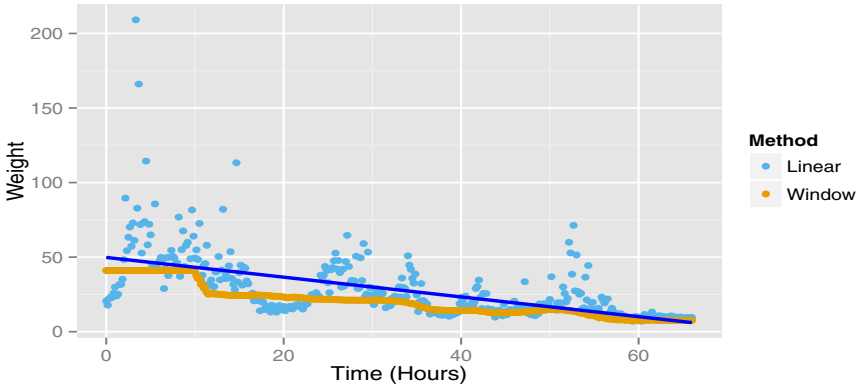


Fig. 3. The weights calculated for each individual time point and then fit with least squares linear regression (blue) to provide weights at each array time point or calculated using a sliding window (gold). Both approaches were compared to the traditional strategy of equal weighting.

Table 2. AUC of coexpression networks built from each U2OS time-course experiment for each gold standard. Three time-course experiments were performed using thymidine-thymidine block, and one was performed using a thymidine-nocadozole block. The weighted methods (Window or Linear) both perform better than the unweighted approach across all datasets and standards.

		Gold Standard			
		Cell Cycle		Cell Cycle & Cycling Expression	
Time-course	Weighting Method	AUC	$\%S_R$	AUC	$\%S_R$
Thy-Thy1	Equal	0.9386	–	1.0000	–
	Window	0.9441	8.96	1.0000	–
	Linear	0.9427	6.68	1.0000	–
Thy-Thy2	Equal	0.9090	–	0.9990	–
	Window	0.9176	9.45	1.0000	100.00
	Linear	0.9175	9.34	1.0000	100.00
Thy-Thy3	Equal	0.8745	–	0.9406	–
	Window	0.8991	19.60	0.9802	66.67
	Linear	0.8936	15.22	0.9754	58.59
Thy-Noc	Equal	0.9132	–	0.9836	–
	Window	0.9299	19.24	0.9850	8.54
	Linear	0.9310	20.51	0.9894	35.37

Table 3. AUC of coexpression networks built from each HeLa time-course experiment for each gold standard. Two time-course experiments were performed using thymidine-thymidine block, and one was performed using a thymidine-nocadozole block. The weighted methods (Window or Linear) both perform better than the unweighted approach across all datasets and standards.

		Gold Standard			
		Cell Cycle		Cell Cycle & Cycling Expression	
Time-course	Weighting Method	AUC	$\%S_R$	AUC	$\%S_R$
Thy-Thy2	Equal	0.8972	–	0.9954	–
	Window	0.8987	1.46	0.9972	39.13
	Linear	0.8974	0.19	0.9972	39.13
Thy-Thy3	Equal	0.9238	–	0.9923	–
	Window	0.9303	8.53	0.9957	44.16
	Linear	0.9284	6.04	0.9957	44.16
Thy-Noc	Equal	0.8238	–	0.9710	–
	Window	0.8310	4.09	0.9850	48.28
	Linear	0.8423	10.50	0.9860	51.72

4 Discussion and Conclusions

We introduce methods using weighted Pearson correlation to better summarize time-course gene expression data. These methods adjust weighting to take into account the loss of synchronization in such experiments. Integrative approaches with such weighted methods are expected to provide a richer set of predicted networks than approaches which ignore this information.

Our results from the HeLa experiment show that the weighting approach can be applied generally. This confirms what we would predict, i.e. that a portion of the observed loss of synchrony is a feature of the experimental method and not of the cell line being assayed. This property allows us to use a weighting function derived from measurements of one cell line (U2OS) to more effectively build coexpression networks from data generated in another cell line (HeLa). This means that we may be able to apply this method directly to other past experiments for which no experimental measures of synchrony are available.

Future work will focus on the automatic identification of time-course or cyclic data, the effective estimation of appropriate weights from the data, and strategies that more effectively leverage such data to build a systems-level portrait of molecular interactions. This work shows that approaches that effectively leverage the relationship between measurements can more effectively build single-experiment coexpression networks. More effective single-dataset networks will provide the groundwork for integrative systems biology methods capable of answering very detailed questions about individual genes and their interactions.

Inexpensive next-generation sequencing methods are expected to make genome-scale assays a routine part of clinical care. Developing approaches that effectively acknowledge and use the temporal relationships within such data will improve our ability to understand an individual's changes in health at the molecular-systems level and thus help to lay the groundwork for precision medicine.

Acknowledgments. Funding was provided via startup funds from The Geisel School of Medicine at Dartmouth to CSG, and National Institutes of Health Grants R01 CA130795, and R01 HG004499 to MLW.

References

1. Troyanskaya, O.G., Dolinski, K., Owen, A.B., Altman, R.B., Botstein, D.: A Bayesian framework for combining heterogeneous data sources for gene function prediction (in Saccharomyces cerevisiae). Proceedings of the National Academy of Sciences of the United States of America 100(14), 8348–8353 (2003)
2. Zhang, Z., Gerstein, M.: Reconstructing genetic networks in yeast. Nature Biotechnology 21(11), 1295–1297 (2003)
3. Lee, I., Date, S.V., Adai, A.T., Marcotte, E.M.: A probabilistic functional network of yeast genes. Science 306(5701), 1555–1558 (2004)
4. Myers, C.L., Troyanskaya, O.G.: Context-sensitive data integration and prediction of biological networks. Bioinformatics 23(17), 2322–2330 (2007)
5. Huttenhower, C., Haley, E.M., Hibbs, M.A., Dumeaux, V., Barrett, D.R., Coller, H.A., Troyanskaya, O.G.: Exploring the human genome with functional maps. Genome Research 19(6), 1093–1106 (2009)
6. Hess, D.C., Myers, C.L., Huttenhower, C., Hibbs, M.A., Hayes, A.P., Paw, J., Clore, J.J., Mendoza, R.M., Luis, B.S., Nislow, C., Giaever, G., Costanzo, M., Troyanskaya, O.G., Caudy, A.A.: Computationally driven, quantitative experiments discover genes required for mitochondrial biogenesis. PLoS Genetics 5(3), e1000407 (2009)
7. Hibbs, M.A., Myers, C.L., Huttenhower, C., Hess, D.C., Li, K., Caudy, A.A., Troyanskaya, O.G.: Directing experimental biology: a case study in mitochondrial biogenesis. PLoS Computational Biology 5(3), e1000322 (2009)
8. Wong, A.K., Park, C.Y., Greene, C.S., Bongo, L.A., Guan, Y., Troyanskaya, O.G.: IMP: a multi-species functional genomics portal for integration, visualization and prediction of protein functions and networks. Nucleic Acids Research 40(Web Server issue), W484–W490 (2012)
9. IMP: Integrative multi-species prediction (October 2012), `http://imp.princeton.edu/networks/data/`
10. Bar-Joseph, Z., Siegfried, Z., Brandeis, M., Brors, B., Lu, Y., Eils, R., Dynlacht, B.D., Simon, I.: Genome-wide transcriptional analysis of the human cell cycle identifies genes differentially regulated in normal and cancer cells. Proceedings of the National Academy of Sciences of the United States of America 105(3), 955–960 (2008)
11. Cho, R.J., Huang, M., Campbell, M.J., Dong, H., Steinmetz, L., Sapinoso, L., Hampton, G., Elledge, S.J., Davis, R.W., Lockhart, D.J.: Transcriptional regulation and function during the human cell cycle. Nature Genetics 27(1), 48–54 (2001)

12. Sadasivam, S., Duan, S., DeCaprio, J.A.: The MuvB complex sequentially recruits B-Myb and FoxM1 to promote mitotic gene expression. Genes & Development 26(5), 474–489 (2012)
13. Whitfield, M.L., Sherlock, G., Saldanha, A.J., Murray, J.I., Ball, C.A., Alexander, K.E., Matese, J.C., Perou, C.M., Hurt, M.M., Brown, P.O., Botstein, D.: Identification of genes periodically expressed in the human cell cycle and their expression in tumors. Molecular Biology of the Cell 13(6), 1977–2000 (2002)
14. Grant, G.D., Gamsby, J., Martyanov, V., Brooks, L., George, L.K., Mahoney, J.M., Loros, J.J., Dunlap, J.C., Whitfield, M.L.: Live-cell monitoring of periodic gene expression in synchronous human cells identifies Forkhead genes involved in cell cycle control. Molecular Biology of the Cell 23(16), 3079–3093 (2012)
15. Yeom, M., Pendergast, J.S., Ohmiya, Y., Yamazaki, S.: Circadian-independent cell mitosis in immortalized fibroblasts. Proceedings of the National Academy of Sciences of the United States of America 107(21), 9665–9670 (2010)
16. Nowrousian, M., Duffield, G.E., Loros, J.J., Dunlap, J.C.: The frequency gene is required for temperature-dependent regulation of many clock-controlled genes in Neurospora crassa. Genetics 164(3), 923–933 (2003)
17. Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D., Futcher, B.: Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. Molecular Biology of the Cell 9(12), 3273–3297 (1998)
18. Langmead, C.J., Yan, A.K., McClung, C.R., Donald, B.R.: Phase-independent rhythmic analysis of genome-wide expression patterns. Journal of Computational Biology: A Journal of Computational Molecular Cell Biology 10(3-4), 521–536 (2003)
19. Johansson, D., Lindgren, P., Berglund, A.: A multivariate approach applied to microarray data for identification of genes with cell cycle-coupled transcription. Bioinformatics 19(4), 467–473 (2003)
20. Wichert, S., Fokianos, K., Strimmer, K.: Identifying periodically expressed transcripts in microarray time series data. Bioinformatics 20(1), 5–20 (2003)
21. Straume, M.: DNA microarray time series analysis: automated statistical assessment of circadian rhythms in gene expression patterning. Methods in Enzymology 383, 149–166 (2004)
22. Chen, J.: Identification of significant periodic genes in microarray gene expression data. BMC Bioinformatics 16(1), 286 (2005)
23. Fan, X., Pyne, S., Liu, J.S.: Bayesian meta-analysis for identifying periodically expressed genes in fission yeast cell cycle. The Annals of Applied Statistics 4(2), 988–1013 (2010)
24. Johnson, D.G., Ohtani, K., Nevins, J.R.: Autoregulatory control of E2F1 expression in response to positive and negative regulators of cell cycle progression. Genes & Development 8(13), 1514–1525 (1994)
25. Alibés, A., Yankilevich, P., Cañada, A., Díaz-Uriarte, R.: IDconverter and IDClight: conversion and annotation of gene and protein IDs. BMC Bioinformatics 8(1), 9 (2007)
26. Myers, C.L., Barrett, D.R., Hibbs, M.A., Huttenhower, C., Troyanskaya, O.G.: Finding function: evaluation methods for functional genomic data. BMC Genomics 7, 187 (2006)
27. Huttenhower, C., Schroeder, M., Chikina, M.D., Troyanskaya, O.G.: The Sleipnir library for computational functional genomics. Bioinformatics 24(13), 1559–1561 (2008)

Inferring Human Phenotype Networks from Genome-Wide Genetic Associations

Christian Darabos[1,⋆], Kinjal Desai[1,*], Richard Cowper-Sal·lari[1], Mario Giacobini[2], Britney E. Graham[1], Mathieu Lupien[3], and Jason H. Moore[1]

[1] Department of Genetics, The Geisel Medical School at Dartmouth College, Lebanon, NH 03756, U.S.A.
[2] Computational Epidemiology Group, Department of Veterinary Sciences, and Complex Systems Unit, Molecular Biotechnology Center, University of Torino, Italy
[3] Ontario Cancer Institute, Princess Margaret Cancer Center-University Health Network, Ontario Institute for Cancer Research and the Department of Medical Biophysics, University of Toronto, Toronto, ON, Canada

Abstract. Networks are commonly used to represent and analyze large and complex systems of interacting elements. We build a human phenotype network (HPN) of over 600 physical attributes, diseases, and behavioral traits; based on more than 6,000 genetic variants (SNPs) from Genome-Wide Association Studies data. Using phenotype-to-SNP associations, and HapMap project data, we link traits based on the common patterns of human genetic variations, expanding previous studies from a gene-centric approach to that of shared risk-variants. The resulting network has a heavily right-skewed degree distribution, placing it in the *scale-free* region of the network topologies spectrum. Additional network metrics hint that the HPN shares properties with *social networks*. Using a standard community detection algorithm, we construct *phenotype modules* of similar traits without applying expert biological knowledge. These modules can be assimilated to the disease classes. However, we are able to classify phenotypes according to shared biology, and not arbitrary disease classes. We present a collection of documented clinical connections supported by the network. Furthermore, we highlight phenotypes modules and links that may underlie yet undiscovered genetic interactions. Despite its simplicity and current limitations the HPN shows tremendous potential to become a useful tool both in the unveiling of the diseases' common biology, and in the elaboration of diagnosis and treatments.

1 Introduction

Biology at the system's level is a holistic approach to the study of an organism's entire phenotypes. When applied to humans, systems biology encompasses all aspects, both environmental and internal, of an individual to understand its traits and diseases. It offers the promise of personalized diagnostics, prognostics and medical treatments [14]. Because of the sheer complexity and the number of interactions, the preferred visualization method of systems biology is the network. Indeed, networks offer relatively straight

⋆ First Co-authors.

L. Vanneschi, W.S. Bush, and M. Giacobini (Eds.): EvoBIO 2013, LNCS 7833, pp. 23–34, 2013.

forward and intuitive representations of interaction phenomena, and allow sophisticated statistical analysis of their intrinsic properties.

In this work, we focus on the system-wide relationships between human phenotypic traits (PT), encompassing physical attributes (e.g. eye color, waist circumference), diseases (e.g. coronary heart disease, Type 1 and 2 diabetes), and behavioral characteristics (e.g. smoking behavior). Elucidating relationships between human traits or diseases is becoming increasingly important in the study of complex genetic disorders. These traits are related through shared genes, proteins and possibly regulatory elements. Identifying these links may help reveal shared mechanisms driving the set of connected diseases. Ultimately, a thorough understanding of these connections may provide the clinical tools necessary to design common drug targets. The potential biological and clinical outcomes justifies efforts to study the phenotype genotypical interactions. These interactions are mathematically and visually represented as a graph: the Human Phenotype Network (HPN). Previous network-based studies of diseases have proven useful for visualizing large disease datasets grouped by common mutated genes, similar gene expression profiles or shared protein interactions [7,24,3]. However, a gene-centric focus has biased the generation and interpretation of these networks, given that coding regions constitute less than 2% of the entire human genome. Genome Wide Association Studies (GWAS) have identified genetic predispositions to disease using a non-candidate-driven approach. To date, approximately 6,000 single nucleotide polymorphisms (SNPs) have been reported as genetic risk-variants for about 600 diseases and traits. Over 90% of risk-associated SNPs (raSNPs) identified by the GWAS fall outside of coding regions ([8]), stressing the requirement for a more global assessment of shared risk-variants. Here we propose a non-gene centric method, relying on genetic risk factors, such as SNPS, and construct a network of traits and diseases based on their shared GWAS loci. Previous studies also focused on classifying diseases into arbitrary disease classes usually based on the organs or the physical location of the disease in the human body, disregarding the shared biology of the diseases. Here, we take a different approach, classifying phenotypes into modules, by using a community detection algorithm based on the phenotypes' position with respect to one another within the HPN. We present a collection of clinical interactions that are corroborated by the network (Fig. 1) and we show that the HPN reveals phenotypes sharing loci that may underlie as yet uncharacterized interactions.

2 Background

In this section, we define the fundamental concepts used in the methods section to build the HPN (Section 3);

2.1 Genome-Wide Association Studies

Genome-wide association studies (GWAS) identify common genetic variants, such as single-nucleotide polymorphisms (SNP), found in the genotype of different individuals in association with phenotypical traits. A SNP is said to be associated with a trait if it is more prevalent in the group presenting the phenotype of interest (cases), when compared to the group not presenting it (controls). SNPs associated with a trait, or risk-associated SNPs (raSNPs), mark the region of the human genome that is believed to

influence the probability (or risk) of the trait's occurrence in an individual [15]. Pairs or groups of SNPs are said to be in *linkage disequilibrium* when they are found to occur together more (or less) often than would be expected at random [6]. The catalog of published GWAS maintained by the National Human Genome Research Institute (NHGRI) at the National Institute of Health (`http://www.genome.gov/gwastudies/`) aggregates studies that report phenotype-to-raSNP(s) associations. The NHGR catalog used in this study, dated 05/17/11, and our primary source of raSNP-trait association data, reports over 600 PTs associated with approximately 6,000 raSNPs.

Imputed Risk Associated Variome. For each trait in the catalog, we extract the complete set of raSNPs, which we call a risk-associated variome (RAV). To address the low genomic coverage provided by GWAS, we associate each raSNP with all SNPs found in linkage disequilibrium (ldSNPs) [6] using the HapMap project data [9]. SNPs in linkage disequilibrium form clusters of variants that statistically appear in the same patient. The HapMap project aims at building a repository of describing the common patterns found in human genetic variations (`http://hapmap.ncbi.nlm.nih.gov/`). The resulting imputed variome (iRAV) will allow us to establish connections between diseases/traits that share blocks, i.e. that have overlapping iRAVs. A recent study [25] shows that SNPs in linkage disequilibrium (ldSNPs) with prostate cancer risk-associated SNPs modulate the expression of an oncogene by altering transcription factor binding sites. The inclusion of ldSNPs in our analysis is therefore expected to be valuable.

2.2 Networks

As previously mentioned, network theory can provide powerful tools for visualizing complex systems. Networks are being used with increasing frequency to analyze large scale systems, such as the Human Disease Network (HDN), which will be the focus of this study. A network can take an extraordinarily complex system and reduce it to a relatively simple form, revealing underlying connections and important clustering details that ordinarily would not be seen, when studying individual or non-complex relationships between traits [16]. Intuitively, a network is a collection of nodes and the edges connecting them. The degree of a node is determined by the number of edges that are attached to it [16]. The degree distribution of a network defines the probability that each node will have a certain degree. The plot of the degree distribution probability function informs us of important global properties of the network. For instance, if the plot curve follows a normal or a Poisson distribution, then the network's topology is said to be *random*. On the other hand, if the plot is right-skewed with a long tail, this indicates that most of the nodes in the network are of a low degree with a few highly connected nodes that are referred to as hubs. This type of network is called *scale-free* and its degree distribution tends to follow a power-law, or decaying exponential curve. Most biological networks are found to be in the scale-free family. When the degree distribution of a scale-free network is plotted on a logarithmic scale, the resulting curve is approximately linear across the top [16]. In the case of relatively small networks, it is impossible to affirm the presence of a scale-free network. We can, at best, show the existence of a power-law type degree distribution, and not dismiss the scale-free hypothesis.

Modules within a Nework. The clustering coefficient (CC) of a network measures the degree to which nodes tend to form closely knit communities with a higher than average connectivity [23]. The CC of networks found in nature, in particular social and biological networks, show a higher degree of clustering than that observed in randomized networks of identical size. This measurement allows one to identify clusters of nodes within the network that are likely to share common attributes based on the structural properties of the network, without using any specific information about the nature of the nodes themselves. These clusters are called modules in the case of general networks, and communities in social networks. The Louvain method of community detection in large scale networks, based on a greedy optimization method [5], is a widely accepted algorithm to build communities (or modules) within a network with no expert-knowledge.

2.3 Human Disease Networks

In recent years there has been a trend toward studying disease through network based analysis of various systems of connections between diseases. The result is the Human Disease Network (HDN). The nodes in the HDN represent human genetic disorders and the edges represent various connections between disorders, such as gene-gene or protein-protein interactions, to name only a few. The HDN is helpful in visualizing human disorders and their corresponding interactions on a large scale, which gives us the opportunity to see the relationships between disorders. The underlying connections of the HDN contribute to the understanding of the basis of disorders, which in turn leads to a better understanding of human diseases.

One study by Goh, *et al.*[7], explored the HDN built on mutated genes shared by different diseases. Another study, which is similar in some ways to ours, by Li *et al.*[13] traced the raSNPs connecting disease traits. In 2009, Silpa Suthram *et al.*[21] found that when diseases were compared and contrasted by an analysis of disease-related messenger RNA (mRNA) expression data and the human protein interaction network, there were significant similarities between certain diseases and that some of the correlated diseases shared drug treatments, as well. This could help us target certain genes for treatment. In 2009, Barrenas *et al.*[3] further studied genetic architecture of complex diseases, by doing a GWAS, and found that complex disease genes are less central than the essential and monogenic disease genes in the human interactome. In the present work, we expand our study to include not only disease traits, but also behaviors and normal variations in humans, such as hair color, and explore large portions of non-coding variations in the human genome. In addition, we include not only raSNPs, but also ldSNPs to achieve a better coverage of the phenotype interactions.

3 HPN Based on Genetic Variations

This section describes our proposed method to construct a HPN of PTs and diseases based on their shared GWAS loci. This model includes data from hundreds of GWAS studies, all catalogued by the NHGRI, and adds the HapMap project data to build comprehensive clusters of rare variants for each phenotype (iRAVs). As we will show in

Section 4, this approach offers interesting insight into the way phenotypes may be linked by common genetic variations. The network is built following the steps below; each step of this algorithm is represented in Fig. 1 below.

1. from the NHGR catalog, extract all PTs associated with at least one raSNP in at least one study, and set those as nodes;
2. associate each PT with the RAV containing all the raSNPs identified;
3. extend each RAV to its iRAV by building clusters of ldSNPs around each raSNP;
4. identify overlapping iRAVs, and connect the associated PTs in the HPN with a directed edge;
5. set the edge weight as the normalized number of iRAVs shared by the 2 PTs, i.e. the number of overlapping iRAVs over the total number of iRAVs associated with the source vertex of that edge;

As a result of Step (1), the network will not contain any isolated nodes. We are only interested in PTs that have been associated with raSNPs, and their possible shared biology. The original NHGR database contains 646 PTs; by removing the isolate nodes, the HPN contains 401 nodes connected to at least 1 other node.

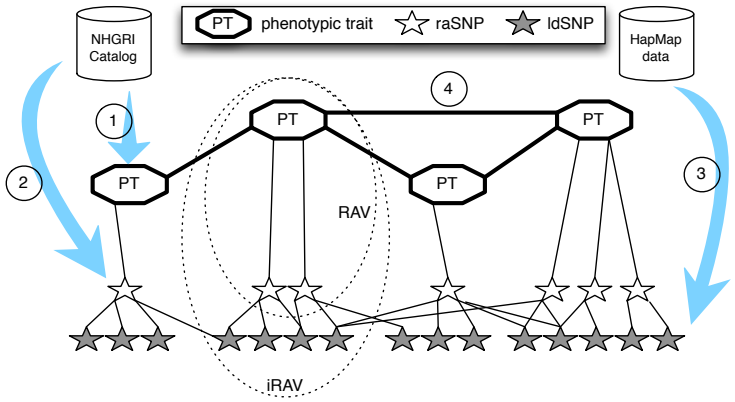


Fig. 1. Step-by-step description of the method to obtain the HPN. The circled numbers correspond to the steps of the method described above.

The resulting network is shown in Fig. 2, where the nodes represent the PTs, and the edges correspond to overlapping iRAVs. To increase the readability, nodes and edges were filtered (see legend of Fig. 2). All the statistics below are, however, computed on the complete (unfiltered) network.

In Fig. 2, the nodes and labels sizes are proportional to the original degree of the PT (before filtering). The edge width is in turn proportional to the number of SNP clusters overlapping in the nodes' iRAVs (weight). Visually, we notice the network is composed of a small number of highly connected hubs: coronary heart disease, cholesterol, Crohn's disease. The vast majority of the nodes are, however, very sparsely connected,

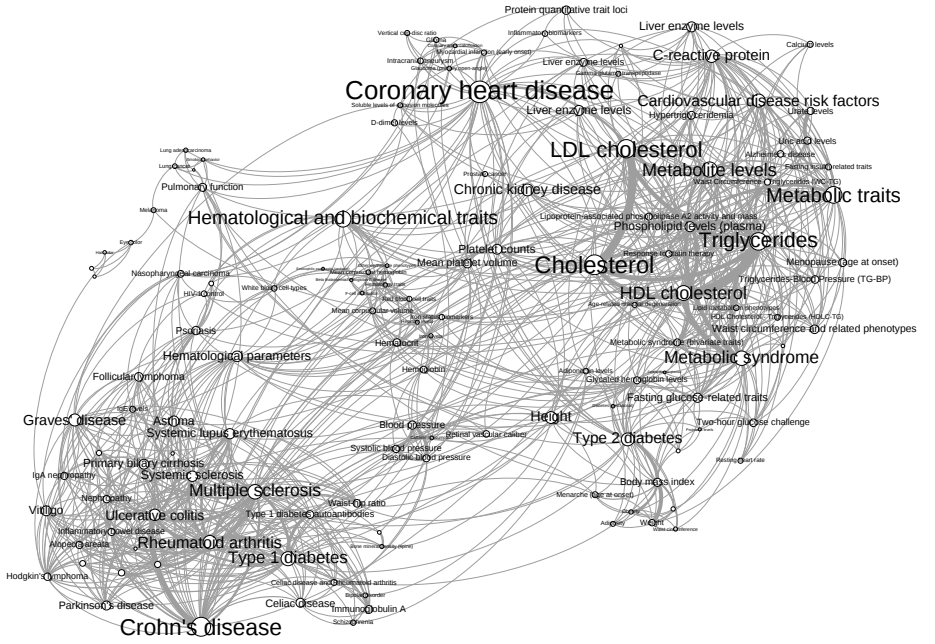


Fig. 2. Human Phenotype Network. In order to increase the readability, we have filtered out nodes with a degree smaller than 5 (i.e. connected to less than 5 other nodes), showing only 137 nodes (about 30%), and edges with a weight lower than 2 (i.e. connecting PTs that have iRAVs overlapping by less than 2 ra/ldSNP clusters), showing about 45% of the actual edges. To further facilitate the readability, we have manually merged a number of clearly redundant nodes and depicted the double directed edges as single undirected.

allowing us to speculate on the scale-free nature of the network. Our hypothesis is supported by the degree distribution plots in Fig. 3. Indeed, the degree distribution is clearly right-skewed, with a *heavy-tail*.

Scale-free networks are ubiquitous in nature and in biology[2], and our HPN is no exception. Crohn's disease and Coronary Heart disease are the main hubs of the network with connections to over 60 other traits. They are followed by Hematological and biochemical traits and LDL Cholesterol related phenotypes. Table 1 summarizes a number of the standard network properties and statistics computed on the HPN. For comparison purposes, we have also included the statistics of the HPN when we disregard ldSNPs, using direct raSNP overlap only. The results are clearly in favor of including ldSNPs into our study, as it offers a more complete view of possible phenotypic interactions.

Indeed, the complete HPN includes about 100 more traits, and about 3 times the number of edges. The HPN's clustering coefficient of 0.595 is much higher than that expected of random networks (RN) of identical dimensions ($CC = 0.035$). The short average path length implies a large number of shortcuts across the network[22]. Moreover,

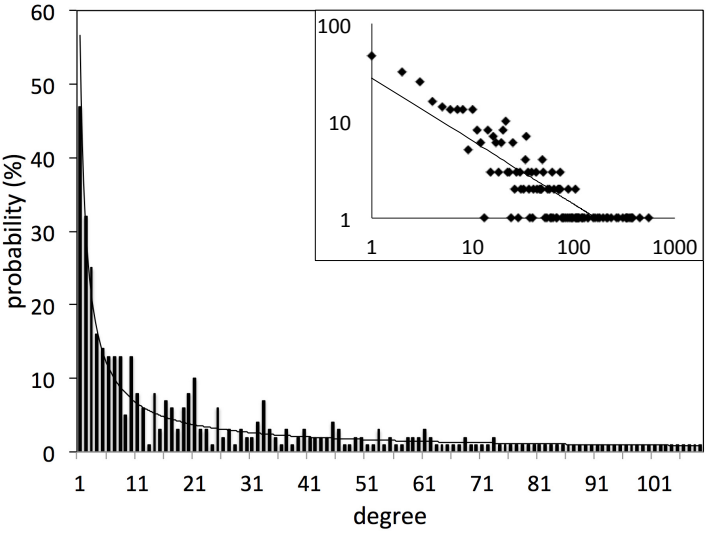


Fig. 3. Degree distribution of the HPN. The vertical axis represents the probability (in percents) of a node having the corresponding degree on the horizontal axis. The inset figure is plotted on a logarithmic scale. The trend lines are shown to offer an approximation of the *long-tailed* function of the distribution and show that the distribution is close to the inverted power-law function of a scale-free distribution.

Table 1. Properties and Statistics of the HPN

Property / Statistic	complete HPN	raSNPs-only HPN
#nodes	401	295
#edges	2845	932
#components	9	25
largest component	385 nodes, 2837 edges	252 nodes, 989 edges
average degree	14.19	6.39
average weighted degree	37.54	10.03
average clustering coefficient	0.595	0.427
average path length	2.961	3.70

the largest connected component (LCC) is significantly smaller than the HPN, where the LCC of a RN with a similar average degree would englobe all nodes. Together, these three properties place the HPN in the *social network* range, where clusters of individuals tend to form with a higher-than-random probability. This also hints that interesting insights can be gained by using a clustering algorithm to identify the HPN's intrinsic communities. In Section 4, we discuss the results of the clustering algorithm (see 2.2), and study the biological and clinical implications that can be gathered from the HPN.

4 Biological and Clinical Implications

In this section, we analyze the constructed HPN and present a collection of clinical interactions that are corroborated by the network (Fig. 4). Furthermore, the network reveals phenotypes sharing loci that may underlie yet uncharacterized interactions. The inclusion of behaviors and traits, as opposed to diseases only, will prove very informative. We use the Louvain method (see Section 2.2) to build the network statistics based communities within our HPN [5]. Fig. 4 offers an overview of the network with a color-coding of the 24 different modules identified by the algorithm. In [7], the authors have manually classified the diseases present in the HDN into "disease classes", leaving an important number "unclassified". Our modules based approach classifies all genetically related phenotypes automatically into approximate classes, based on their linkage within the HPN. This contrasts strongly with previous work, where arbitrary disease classes grouped phenotype regardless of their shared biological attributes. For example, in previous studies, all cancers were part of the same class, regardless of the cancer type. Using the phenotype modules and the community detection algorithm, our framework is able to classifies each cancer with the phenotype that they are most likely to share genetic attributes with. Because of the large differences in the data sets, notably the addition of physical and behavioral traits, we cannot directly compare the classes used in the HDN and our HPN modules. The largest clusters appear around the hubs: "Crohn's disease" and "Type 1 diabetes" in Fig. 4 cluster A; "Coronary heart disease" and "Hematological and biochemical traits", Fig. 4 cluster B; "C-reactive protein" and "Chronic kidney disease", Fig. 4 cluster C; "LDL cholesterol" and "Triglycerides" (metabolic diseases); and "Type 2 diabetes" and "Obesity" (metabolic).

The HPN presents several edges that confirm well-characterized genetic interactions. These include the dense interconnectivity between immune-related disorders and phenotypes. For instance, systemic sclerosis and rheumatoid arthritis, both autoimmune disorders in the systemic inflammatory rheumatic disease family, are connected and are part of the same module (Fig. 5B). The metabolic diseases centered around excessive weight, elevated body mass index, obesity and Type 2 diabetes also form a module (Fig. 5C).

The HDN also points to connections between diseases known to rely on common factors. For instance, bone mineral density (hip and spine) is linked with inflammatory diseases such as Crohns disease and Ulcerative Colitis, both subsets of Inflammatory Bowel Disorder (IBD). Nuclear factor kappa B (NFκB) is known to be involved in driving IBD [10] and has recently been shown to play a role in regulating genes responsible for bone formation [12] a key factor in establishing bone mineral density. Bone mineral density (hip and spine) is also connected with breast cancer (not shown here). It is well established that the estrogen receptor alpha (ESR1) drives oncogenesis in over two-thirds of all breast cancers [17]. Mutations in the estrogen receptor gene have also been associated with loss of bone mineral density in humans [20]. The HDN also reveals connections between behavioral traits and diseases. For instance, lung cancer is connected with smoking behavior and nicotine dependence within the same module (Fig. 5B). This raises the possibility that a SNP associated with a certain disease phenotype may not affect the biology in the disease tissue but instead promote behaviors that increase the risk of the connected disease. The HDN also shows a connection between

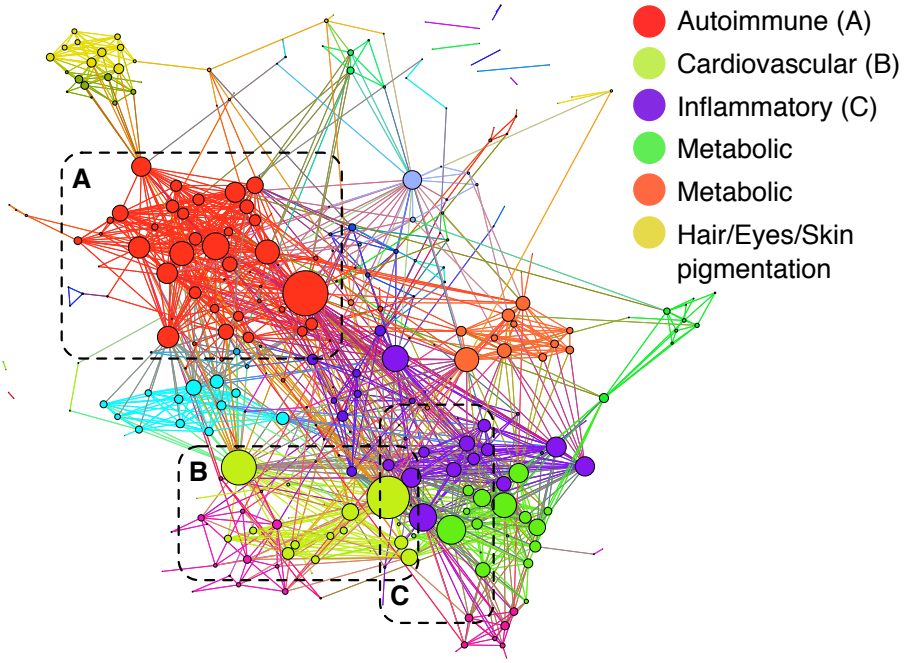


Fig. 4. Network of Phenotypes. Nodes represent phenotypic traits: physical attributes, diseases and behaviors assessed for genetic predisposition through GWAS. Node sizes are proportional to the number of iRAVs associated with the phenotype. The edge weights are based on the number of shared iRAVs normalized by the number of iRAVs associated with the source node. Modules are calculated based on the Louvain method of greedy optimization. Several intuitive modules occur, such as immunological traits or hair and skin pigmentation clusters. The most important modules are shown according to the legend. Chosen modules A, B, and C will be focused on in Fig. 5.

lung cancer and systemic lupus erythematosus (Figs. 5B&C). Consistent with this, it has been shown that lupus patients show an increased risk of lung cancer [4]. Interestingly, the over-the-counter drug cimetidine, found to significantly decrease the lung adenocarcinoma tumor burden compared to untreated controls in mice [19], has also been administered to patients who develop lupus nephritis, an inflammation of the kidney caused by systemic lupus erythematosus, to improve renal function [18]. Finally, the HDN may help uncover biomarkers for diseases. For example, a connection between C-reactive protein (CRP), whose levels in the blood rise in response to inflammation, and Alzheimers disease (AD) is noted (Fig. 5C). Clinically, plasma levels of CRP remain normal in AD patients. However, specific polymorphisms in the regulatory regions of the CRP gene are associated with patients at risk of developing AD [11], different from the SNP shared by the two disease phenotypes. Our results show the implications of disease connectivity, via shared risk-loci, which should help to better understand the shared etiology of linked diseases. This is highlighted by the connection between CRP and AD. CRP was recently found to be a biomarker for AD even though NFκB has

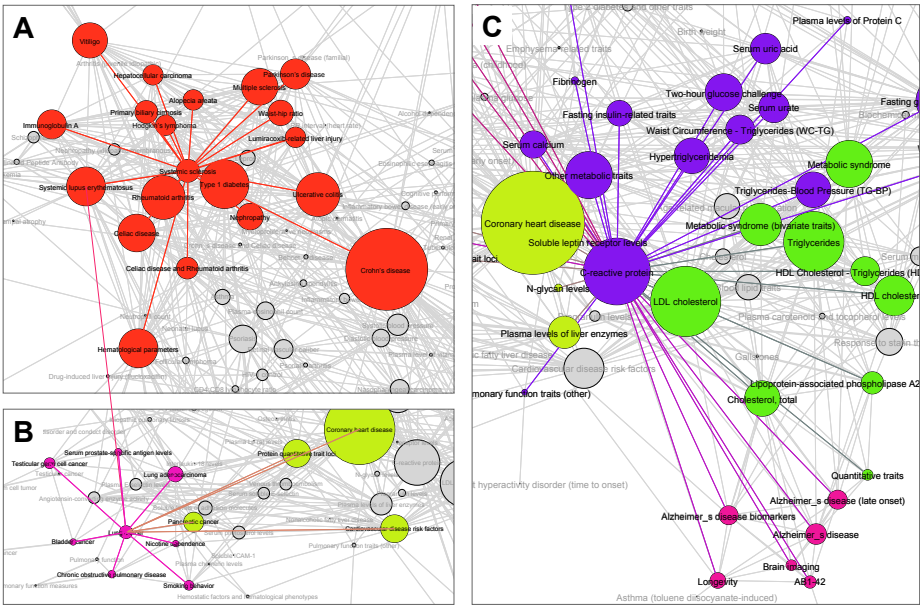


Fig. 5. Network of Phenotypes: In Panel A, traits associated with systemic sclerosis. The connections reflect established associations of systemic sclerosis to immunological diseases and traits clustered within the same module. In Panel B, traits associated with lung cancer. The connection between lung cancer and systemic lupus erythematosus reflects a higher disease risk of lung cancer in cohorts of lupus patients which was only discovered incidentally by previous studies. In Panel C, traits associated with the C reactive protein (CRP). Previous clinical studies have linked CRP to diseases like coronary heart disease and Alzheimers disease.

been long known to play a role in AD development [1]. A connection between the two, however, is seen readily in our network. Additionally, the wealth of known clinical interactions between diseases and traits can be mapped to the risk-associated loci that they share. Hence, this phenotypic network based on genetic predispositions will likely lead to focused biological studies of diseases based on their unique or shared genetic predispositions.

5 Discussion and Future Work

In this study, we successfully build a network of human traits based on about 6,000 genetic variants of over 600 phenotypical traits. We obtain a network that is more comprehensive than that of previous studies by combining GWAS data, which associates PT to raSNPs, with the HapMap project, which links raSNPs to ldSNPs. Our statistical analysis of the network properties places the HPN in the *scale-free* family, showing once more how ubiquitous network structures with heavy-tailed degree distributions really are in the biological, social, and natural networks in general. The automatic classification of phenotypes into “phenotype classes”, using the networks topological modularity

and a standard community detection algorithm, showed very promising results. Indeed, in contrast to what was achieved in previous studies and manual classification, we are able to highlight modules with phenotypes with potentially interesting shared biology, not by arbitrary disease types (i.e. all cancers are classified together regardless of their genetic background). Despite its simplicity, the HPN both confirmed the existence of commonly known phenotype interactions, and also unveiled links that have nevertheless been characterized in recent literature. Because of these findings, we are highly confident that the HPN, and its subsequent revisions, has the potential to become an advantageous clinical tool, both in helping to discover shared biology between PTs, and for possible development of common target drugs. We are currently working on using statistical methods to help us filter out the connections that are genetically and statistically less probable. We would also like to include different datasets, and analyze the overlap of a HDN using genes, pathways, and protein interactions.

Acknowledgments. This work was partially supported by NIH grants R01 EY022300, LM009012, LM010098, AI59694, by the Swiss National Science Foundation grant PBLAP3-136923, by the New Hampshire IDeA Network of Biological Research Excellence (NH-INBRE) with grants from the National Center for Research Resources (5P20RR030360-03) and the National Institute of General Medical Sciences (8P20GM103506-03), and by the Joint Research Program 2011 of the University of Torino and the Compagnia di San Paolo.

References

1. Akiyama, H., Barger, S., Barnum, S., Bradt, B., Bauer, J., Cole, G.M., Cooper, N.R., Eikelenboom, P., Emmerling, M., Fiebich, B.L., Finch, C.E., Frautschy, S., Griffin, W., Hampel, H., Hull, M., Landreth, G., Lue, L.-F., Mrak, R., Mackenzie, I.R., McGeer, P.L., O'Banion, M., Pachter, J., Pasinetti, G., Plata-Salaman, C., Rogers, J., Rydel, R., Shen, Y., Streit, W., Strohmeyer, R., Tooyoma, I., Muiswinkel, F.L.V., Veerhuis, R., Walker, D., Webster, S., Wegrzyniak, B., Wenk, G., Wyss-Coray, T.: Inflammation and alzheimer's disease. Neurobiology of Aging 21(3), 383–421 (2000)
2. Albert, R.: Scale-free networks in cell biology. J. of Cell Science 118, 4947–4957 (2005)
3. Barrenas, F., Chavali, S., Holme, P., Mobini, R., Benson, M.: Network properties of complex human disease genes identified through genome-wide association studies. PLoS ONE 4(11), e8090 (2009)
4. Bin, J., Bernatsky, S., Gordon, C., Boivin, J.-F., Ginzler, E., Gladman, D., Fortin, P.R., Urowitz, M., Manzi, S., Isenberg, D., Rahman, A., Petri, M., Nived, O., Sturfeldt, G., Ramsey-Goldman, R., Clarke, A.E.: Lung cancer in systemic lupus erythematosus. Lung Cancer 56(3), 303–306 (2007)
5. Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment 2008(10), P10008+ (2008)
6. Falconer, D.S., Mackay, T.F.C.: Introduction to Quantitative Genetics, 4th edn. Prentice Hall (February 1996)
7. Goh, K.-I., Cusick, M.E., Valle, D., Childs, B., Vidal, M., Barabasi, A.-L.: The human disease network. Proceedings of the National Academy of Sciences 104(21), 8685–8690 (2007)

8. 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)
9. The International HapMap Consortium: A haplotype map of the human genome. Nature 437(7063), 1299–1320 (2005)
10. Jobin, C., Sartor, R.B.: Nf-kappab signaling proteins as therapeutic targets for inflammatory bowel diseases. Inflamm. Bowel Dis. 6(3), 206–213 (2000)
11. Kok, E.H., Alanne-Kinnunen, M., Isotalo, K., Luoto, T., Haikonen, S., Goebeler, S., Perola, M., Hurme, M.A., Haapasalo, H., Karhunen, P.J.: Crp gene variation affects early development of alzheimer's disease-related plaques. J. Neuroinflammation 8, 96 (2011)
12. Krum, S.A., Chang, J., Miranda-Carboni, G., Wang, C.-Y.: Novel functions for nfkappab: inhibition of bone formation. Nat. Rev. Rheumatol. 6(10), 607–611 (2010)
13. Li, H., Lee, Y., Chen, J.L., Rebman, E., Li, J., Lussier, Y.A.: Complex-disease networks of trait-associated single-nucleotide polymorphisms (SNPs) unveiled by information theory. Journal of the American Medical Informatics Association: JAMIA 19(2), 295–305 (2012)
14. Loscalzo, J., Barabasi, A.-L.: Systems biology and the future of medicine. Wiley Interdisciplinary Reviews: Systems Biology and Medicine 3(6), 619–627 (2011)
15. McCarthy, M.I., Abecasis, G.R., Cardon, L.R., Goldstein, D.B., Little, J., Ioannidis, J.P.A., Hirschhorn, J.N.: Genome-wide association studies for complex traits: consensus, uncertainty and challenges. Nat. Rev. Genet. 9(5), 356–369 (2008)
16. Newman, M.: Networks: An Introduction. Oxford University Press, Inc., New York (2010)
17. Osborne, C.K.: Steroid hormone receptors in breast cancer management. Breast Cancer Res. Treat. 51(3), 227–238 (1998)
18. Roubenoff, R., Drew, H., Moyer, M., Petri, M., Whiting-O'Keefe, Q., Hellmann, D.B.: Oral cimetidine improves the accuracy and precision of creatinine clearance in lupus nephritis. Ann. Intern. Med. 113(7), 501–506 (1990)
19. Sirota, M., Dudley, J.T., Kim, J., Chiang, A.P., Morgan, A.A., Sweet-Cordero, A., Sage, J., Butte, A.J.: Discovery and preclinical validation of drug indications using compendia of public gene expression data. Sci. Transl. Med. 3(96), 96–477 (2011)
20. Smith, E.P., Boyd, J., Frank, G.R., Takahashi, H., Cohen, R.M., Specker, B., Williams, T.C., Lubahn, D.B., Korach, K.S.: Estrogen resistance caused by a mutation in the estrogen-receptor gene in a man. N. Engl. J. Med. 331(16), 1056–1061 (1994)
21. Suthram, S., Dudley, J.T., Chiang, A.P., Chen, R., Hastie, T.J., Butte, A.J.: Network-based elucidation of human disease similarities reveals common functional modules enriched for pluripotent drug targets. PLoS Comput. Biol. 6(2), e1000662 (2010)
22. Watts, D.J.: Small Worlds: The Dynamics of Networks Between Order and Randomness. Princeton University Press, Princeton (1999)
23. Watts, D.J., Strogatz, S.H.: Collective dynamics of "small-world" networks. Nature 393, 440–442 (1998)
24. Wu, X., Jiang, R., Zhang, M.Q., Li, S.: Network-based global inference of human disease genes. Mol. Syst. Biol. 4, 189 (2008)
25. Zhang, X., Cowper-Sal lari, R., Bailey, S.D., Moore, J.H., Lupien, M.: Integrative functional genomics identifies an enhancer looping to the sox9 gene disrupted by the 17q24.3 prostate cancer risk locus. Genome Res. 22(8), 1437–1446 (2012)

Knowledge-Constrained K-Medoids Clustering of Regulatory Rare Alleles for Burden Tests

R. Michael Sivley, Alexandra E. Fish, and William S. Bush

Center for Human Genetics Research, Department of Biomedical Informatics,
Vanderbilt University, Nashville, TN, USA
{mike.sivley,alexandra.e.fish,william.s.bush}@vanderbilt.edu

Abstract. Rarely occurring genetic variants are hypothesized to influence human diseases, but statistically associating these rare variants to disease is challenging due to a lack of statistical power in most feasibly sized datasets. Several statistical tests have been developed to either collapse multiple rare variants from a genomic region into a single variable (presence/absence) or to tally the number of rare alleles within a region, relating the burden of rare alleles to disease risk. Both these approaches, however, rely on user-specification of a genomic region to generate these collapsed or burden variables, usually an entire gene. Recent studies indicate that most risk variants for common diseases are found within regulatory regions, not genes. To capture the effect of rare alleles within non-genic regulatory regions for burden tests, we contrast a simple sliding window approach with a knowledge-guided k-medoids clustering method to group rare variants into statistically powerful, biologically meaningful windows. We apply these methods to detect genomic regions that alter expression of nearby genes.

1 Introduction

Numerous studies have been published illustrating the association of commonly occurring genetic variants to traits of interest in humans [1], and to changes in gene expression [2]. Recent technological advances in sequencing technology have enabled the study of rare variation – single base-pair changes in DNA that occur at less than 5% frequency in a population [3]. Typical genetic association studies rely on linear or logistic regression models to contrast the phenotype of interest across genotype categories based on a single variant (i.e. AA [25%], Aa [50%], and aa [25%]). Statistical power for these studies is directly related to the frequencies of these genotype categories, and lower frequency variants often have extremely low power to detect associations using these methods because most individuals in the study do not have the rare variant (i.e. AA [98%], Aa [1.8%], and aa [0.2%]).

Multiple methods have been proposed to address the issues of statistical power [4], all of which rely on grouping rare variants together either by biological function or physical proximity in the genome. The vast majority of these statistical methods provide users with the flexibility to specify the genomic region they wish to use for grouping variants together. In practice, variants are typically collapsed within gene

L. Vanneschi, W.S. Bush, and M. Giacobini (Eds.): EvoBIO 2013, LNCS 7833, pp. 35–42, 2013.

regions under the hypothesis that a variants influence disease by changing coding DNA that impacts protein function in some way. However, recent publications by the ENCODE project have shown that the vast majority of previously identified genetic associations are non-coding and regulatory in nature[5].

Currently, non-genic approaches to group rare variants include a simple sliding window approach [6] or collapsing variants within regions defined by experimental data, such as the ENCODE annotations. Sliding window approaches require millions of statistical tests which are highly correlated. The large number of tests makes determining the false positive or false discovery rate of the analysis challenging. Collapsing variants within putative regulatory regions may produce windows that are too small to capture variants to provide a powerful test. This approach also assumes that the genomic locations of regulatory regions are well-defined – an unlikely assumption for many Chromatin Immuno-Precipiation (ChIP) experiments [7]. Therefore, new methods for defining non-genic windows for statistical analysis are needed.

In this work, we apply k-medoids clustering to leverage both physical proximity and biological function with the goal of defining groups of rare variants for statistical analysis. We use a single source of putative biological function – a prediction of genome function based on chromatin state – and refine groupings using physical proximity in the genome. We apply this clustering method to generate rare variant groupings and evaluate the impact of these grouped variants on gene expression traits. Results from our clustering-based approach are compared with a traditional sliding window approach.

2 Methods

2.1 Data

Publically available datasets with phased haplotype information and whole-genome gene expression data on 1000 Genomes samples were used [3]. There were 149 independent, multi-ethnic individuals, consisting of 32 CEPH (CEU) and 37 Yoruba (YRI) parental samples, and 41 Chinese (CHB) and 39 Japanese (JPT) unrelated individuals. Phased haplotype data was obtained from the imputation reference panels for MaCH software (1000G Phase 1 version 3 MaCH panels) and was based upon 1000 Genomes Phase 1 integrated genotype calls and included singleton variants [8]. For gene expression data, we accessed normalized gene expression data from [2] (available online: http://eqtnminer.sourceforge.net/), which was generated using Illumina human whole-genome expression arrays (WG-6 version 1) on lymphoblastoid cell lines from each of the 149 individuals. Expression data was first normalized by quantile normalization within replicates, and then was median normalized across individuals. Additionally, we applied Gaussian quantile normalization for the test genes within each population, in order to account for population differences in gene expression. This normalization was congruent with the original normalization performed in [2]. For each of the selected genes, we extracted genotypes in the *cis*-regulatory region (500KB upstream of the transcriptional start site and 500KB downstream of the transcriptional end site).

2.2 Domain Knowledge

We used classification results from a published study of chromatin marks [9] to guide our cluster analysis. This study used ChIP data to identify methylation and acetylation modifications to histone proteins throughout the genome for nine cell lines. These patterns form the *histone code* [10], and were classified using a multivariate Hidden Markov Model into 15 states, which we loosely grouped into promoter, enhancer, insulator, and transcribed regions. Because our analysis was focused exclusively on gene expression in lymphoblastoid cell lines, we used chromatin state classifications generated for the GM12878 lymphoblastoid cell line. This data is available via the ENCODE project website through the UCSC genome browser (http://genome.ucsc.edu/ENCODE/). By guiding our cluster analysis with this data, we hypothesize that genetic variation within similar chromatin states should be grouped together.

2.3 Gene Selection

To compare the two methods across a variety of different regulatory architectures, four genes were selected from a group of genes previously identified as having collections of rare variants functioning as cis-eQTLs, based upon a genome-wide collapsing analysis (unpublished data). Each gene selected represents a potentially unique regulatory architecture, based upon the functional annotation of rare variants which were within the significant regions. Rare variants within significant regions could be identified as disrupting a transcription factor binding site (*ORMDL1*), being present in a ChIP peak (*NUDT22*), or having no functional annotation whatsoever (*FAM154B*). A potential confounder to this study is the presence of common eQTLs in significant regions. A compilation of known common eQTLs was used to determine that none of the above genes had a common eQTL in the previously identified significant regions. To interrogate the effects of common eQTLs on the analysis, *DYPSL4* was also selected, which contained three common eQTLs in the previously identified significant region in addition to rare variants affecting transcription factor binding sites.

2.4 Cluster-Based Analysis

Constrained Partitioning (COP) is a method by which partial knowledge can be introduced into a clustering algorithm, making it a semi-supervised method. Constraints allow for otherwise uninformed clustering methods to include background knowledge of a particular domain. Typically, COP is provided with a list of must-link constraints and cannot-link constraints, which dictate which observations must and cannot be placed in the same cluster.

In our implementation, we allow for an initial classification of chromatin state SNPs surrounding a gene. This classification acts as a must-link constraint for all observations in a class, and a cannot-link constraint for all observations of differing classes. We then apply Partitioning Around Medoids (PAM) to subdivide these SNPs

according to their base position. PAM divides the data into k clusters, where k is specified *a priori* [11]. To choose an optimal k, we ran PAM multiple times with increasing k and select k such that it maximizes with average silhouette width of the resultant clusters. The choice of k is made for each initial classification and the original classes do not need to be partitioned into the same number of clusters.

With our rare variants clustered, we then performed a rare variant burden test, which collapses the data into a single variable, indicating the number of rare variants within that cluster. For each cluster, linear regression was used to determine the significance of association between the clustered rare variants and gene expression. This implementation was done entirely in R.

2.5 Sliding Window Analysis

A rare variant burden test with sliding windows was performed on the test genes. For each gene, the region tested consisted of 500KB both up and downstream, in addition to the gene itself. In this region, a 5KB sliding window was used, such that each SNP served as the start point for a window. All rare variants in this 5KB region were used to determine the burden of rare variants. Only windows with at least one rare variant detected were included in analysis. For each window, a linear regression was performed between the number of rare variants present within a region for each individual and the gene expression level. This is slightly different from the analysis used to select the genes, in which individuals were placed into a binary category of either having a rare variant or not – a *collapsing* test [12].

2.6 Determination of Significance

The best practice for the statistical analysis of sliding windows is a current topic of debate. To place these results in the context of standard genetic analysis guidelines, both a Bonferroni correction and a False Discovery Rate (FDR) analysis were performed [13]. Each gene was analyzed independently in both the Bonferroni and FDR (FDR = 0.05) analyses. In the Bonferroni correction analysis, the number of clusters present in each gene is used to set the gene-specific significance threshold for cluster data. For the sliding window analysis, the number of windows set the gene-specific significance threshold. After being identified as significant, all overlapping windows were merged to form a significant 'signal' in the sliding window analysis.

2.7 Visualization

We visualized the results from both the sliding window and cluster analyses in a single plot using the R package ggplot2 [14]. For the sliding window analysis, the midpoint chromosome position of each 5KB window is plotted relative to the $-\log_{10}$ of the regression p-value to generate a *Manhattan* plot. We used loess to fit a smooth curve to these data points using the stat_smooth function with a span parameter of

0.2. Results from the cluster analysis are shown as horizontal bars (to illustrate the span of the cluster) plotted relative to the $-\log_{10}$ of the regression p-value, color coded by chromatin state. Note that some clusters are too small to be seen on these plots.

3 Results

3.1 Gene Region Results

Visual comparisons of sliding window and cluster analysis approaches are provided in figure 1. *ORMDL1* best illustrates the potential of this method. A highly significant effect is seen from an enhancer cluster which overlaps with the strongest effect from the sliding window analysis. *NUDT22* also shows a strong effect of a large enhancer cluster which spans the best sliding window effect. For both these genes the clustering results correlate well with the loess curves, capturing the 'shape' of the regional effect. The cluster analysis shows less utility for *DYPSL4*, a gene with complex common eQTL effects, and *FAM154B,* a gene with no obvious regulatory mechanisms. For these genes, the method clustered together distant variants within insulator elements creating single clusters containing variants at great distances; these clusters do not reflect the domain knowledge well. We plan to refine the algorithm to include additional constraints limiting the physical distance separating rare variants within potential clusters.

3.2 Bonferroni Correction

The summary of significant genomic regions with a Bonferroni corrected analysis is presented in Table 1. Similar numbers of significant genomic regions are returned by both the sliding window and clustering analysis. In both methods, *DYPSL4* failed to result in significant results. In the case of *ORMDL1*, both clustering and sliding window analysis each resulted in one unique significant region which was not overlapping. All other significant regions overlapped with a region identified in the other test. In *NUDT22*, all significant signals identified by sliding window analysis overlapped with significant clusters. Cluster analysis additionally resulted in two unique significant regions. None of the significant regions identified in *FAM154B* overlapped between the sliding window analysis and the clustering analysis.

Table 1. Number of significant genomic regions detected using both clustering and sliding window analysis with a Bonferroni correction for multiple testing

GENE	Bonferroni Threshold for Cluster Analysis	Number of Significant Clusters	Bonferroni Threshold for Sliding Window Analysis	Number of Significant Windows from Sliding Window Analysis
ORMDL1	0.001250	6 of 40	3.95476×10^{-6}	604 of 12,643
NUDT22	0.001351	5 of 37	4.64857×10^{-6}	26 of 10,756
DYPSL4	0.001282	0 of 39	3.16476×10^{-6}	0 of 15,799
FAM154B	0.001351	3 of 37	6.38162×10^{-6}	32 of 7,835

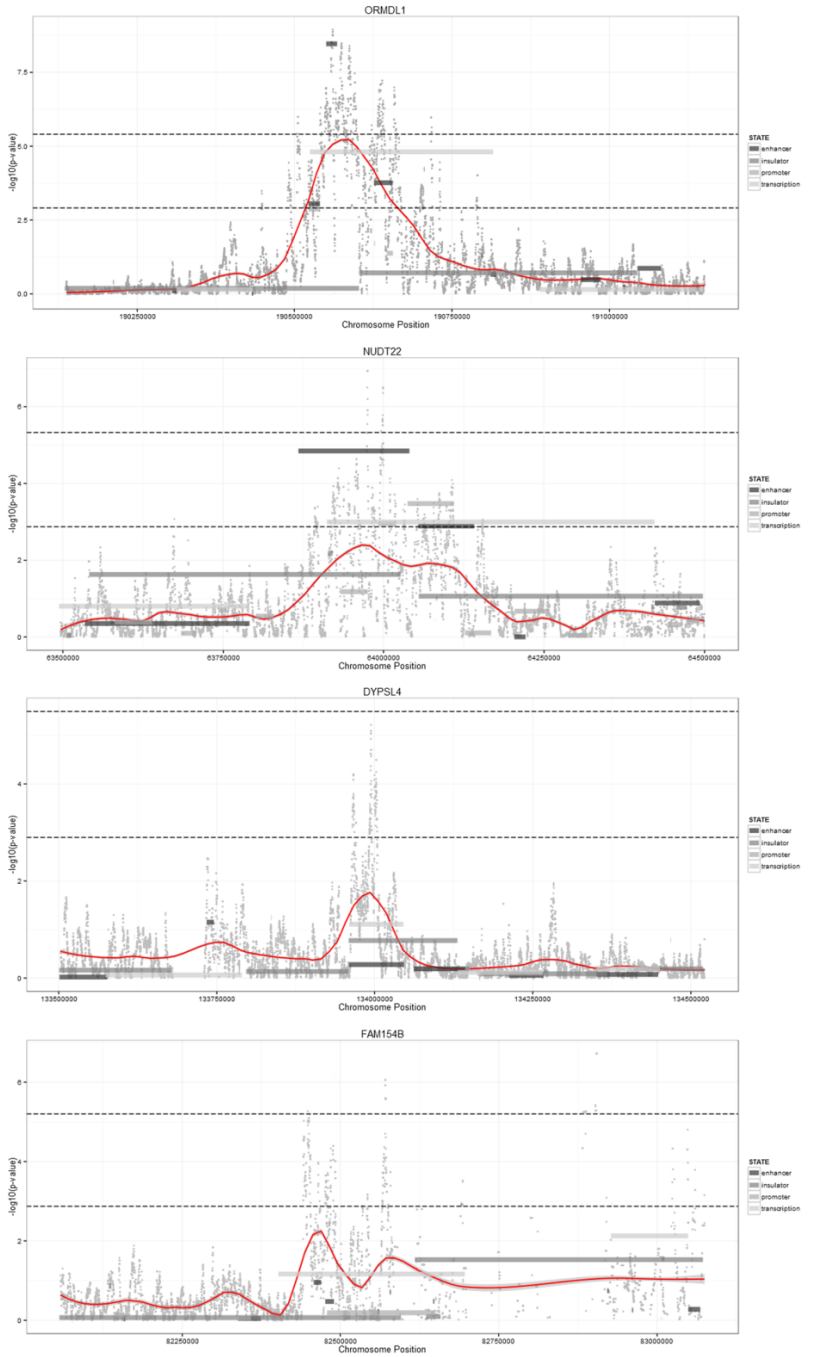


Fig. 1. Manhattan plot of window midpoints (points), variant clusters (bars) by significance with loess fit (red line, loess span = 0.2) of window midpoint by significance

3.3 False Discovery Rate Correction

The significant genomic regions with a FDR (FDR = 0.05) corrected analysis are presented in Table 2. All the regions identified as significant with the Bonferroni correction were identified with the FDR correction as well. One unique cluster was identified with FDR analysis in both *ORMDL1* and *NUDT22*. A dramatic increase was observed in the number of signals identified as significant in the sliding window analysis. For *ORMDL1*, *NUDT22*, and *FAM154B*, all significant clusters overlapped with regions identified as being significant by sliding window analysis. In the case of *DYPSL4*, clustering failed to identify any significant regions, whereas sliding window analysis identified two genomic regions as significant. Sliding window analysis identified a total of 28 unique genomic regions as significant in these genes.

Table 2. Number of significant genomic regions detected using both clustering and sliding window analysis with an FDR=0.05 correction for multiple testing. *There are no p-values < 0.05, making it impossible to calculate the FDR = 0.05 threshold.

GENE	Threshold for Cluster Analysis FDR = 0.05	Number of Significant Clusters	FDR = 0.05 Threshold for Sliding Window Analysis	Number of Significant Windows from Sliding Window Analysis
ORMDL1	0.001346812	7 of 40	0.007989149	2021 of 12,643
NUDT22	0.006583255	6 of 37	0.007619227	1126 of 10,756
DYPSL4	NA*	0 of 39	0.000434797	126 of 15,799
FAM154B	0.001213077	3 of 37	0.006232502	628of 7,835

4 Discussion

Our results indicate that informed clustering of rare variants using regulatory annotations can dramatically reduce the number of statistical tests, reducing the multiple testing burden for rare variant analysis, thus increasing overall power. Obviously, this approach will perform best when the underlying assumption of the method holds true; that influential variants fall within regulatory regions, as illustrated in the *ORMDL1* gene.

A great strength of this approach is that the clustering is independent of statistical analysis, and can be coupled with various methods, such as the Sequence Kernel Association Test(SKAT) or KBAC [15, 16]. Because the method is unsupervised, there are no over-fitting concerns in the association analysis, and standard statistical assumptions of these tests are not violated. The cluster method could also be informed by statistical power calculations of the coupled association test (or other testing assumptions), allowing clusters of rare variants to be optimized to improve the overall power of the analysis. Finally, in this study we have used chromatin state data to guide cluster formation, however numerous other genomic annotations could be applied simultaneously to intelligently design functional clusters of rare variants. As ENCODE and other projects continue to expand our understanding of gene regulation, methods that can leverage this data for analysis will become ever more important.

Acknoweldgements. This work was supported in part by NIH U01 HG004798 and its ARRA supplements.

References

1. 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. Proceedings of the National Academy of Sciences of the United States of America 106, 9362–9367 (2009)
2. Veyrieras, J.-B., Kudaravalli, S., Kim, S.Y., Dermitzakis, E.T., Gilad, Y., Stephens, M., Pritchard, J.K.: High-resolution mapping of expression-QTLs yields insight into human gene regulation. PLoS Genetics 4, e1000214 (2008)
3. Durbin, R.M., Altshuler, D.L., Abecasis, G.R., Bentley, D.R., Chakravarti, A., et al.: A map of human genome variation from population-scale sequencing. Nature 467, 1061–1073 (2010)
4. Bansal, V., Libiger, O., Torkamani, A., Schork, N.J.: Statistical analysis strategies for association studies involving rare variants. Nature Reviews Genetics 11, 773–785 (2010)
5. Schaub, M.A., Boyle, A.P., Kundaje, A., Batzoglou, S., Snyder, M.: Linking disease associations with regulatory information in the human genome. Genome Research 22, 1748–1759 (2012)
6. Lawrence, R., Day-Williams, A.G., Elliott, K.S., Morris, A.P., Zeggini, E.: CCRaVAT and QuTie-enabling analysis of rare variants in large-scale case control and quantitative trait association studies. BMC Bioinformatics 11, 527 (2010)
7. Mendenhall, E.M., Bernstein, B.E.: DNA-protein interactions in high definition. Genome Biology 13, 139 (2012)
8. Li, Y., Willer, C.J., Ding, J., Scheet, P., Abecasis, G.R.: MaCH: using sequence and genotype data to estimate haplotypes and unobserved genotypes. Genetic Epidemiology 34, 816–834 (2010)
9. Ernst, J., Kheradpour, P., Mikkelsen, T.S., Shoresh, N., Ward, L.D., Epstein, C.B., Zhang, X., Wang, L., Issner, R., Coyne, M., Ku, M., Durham, T., Kellis, M., Bernstein, B.E.: Mapping and analysis of chromatin state dynamics in nine human cell types. Nature 473, 43–49 (2011)
10. Rando, O.J.: Combinatorial complexity in chromatin structure and function: revisiting the histone code. Current Opinion in Genetics & Development 22, 148–155 (2012)
11. Kaufman, L., Rousseeuw, P.: Clustering by means of medoids (1987)
12. Li, B., Leal, S.M.: Methods for detecting associations with rare variants for common diseases: application to analysis of sequence data. American Journal of Human Genetics 83, 311–321 (2008)
13. Storey, J.D., Tibshirani, R.: Statistical significance for genomewide studies. Proceedings of the National Academy of Sciences of the United States of America 100, 9440–9445 (2003)
14. Wickham, H.: ggplot2: elegant graphics for data analysis. Springer, New York (2009)
15. Liu, D.J., Leal, S.M.: A novel adaptive method for the analysis of next-generation sequencing data to detect complex trait associations with rare variants due to gene main effects and interactions. PLoS Genetics 6, e1001156 (2010)
16. Wu, M.C., Lee, S., Cai, T., Li, Y., Boehnke, M., Lin, X.: Rare-variant association testing for sequencing data with the sequence kernel association test. American Journal of Human Genetics 89, 82–93 (2011)

Feature Selection and Classification of High Dimensional Mass Spectrometry Data: A Genetic Programming Approach

Soha Ahmed[1], Mengjie Zhang[1], and Lifeng Peng[2]

[1] School of Engineering and Computer Science
[2] School of Biological Sciences
Victoria University of Wellington, PO Box 600, Wellington 6140, New Zealand
{soha.ahmed,mengjie.zhang}@ecs.vuw.ac.nz, lifeng.peng@vuw.ac.nz

Abstract. Biomarker discovery using mass spectrometry (MS) data is very useful in disease detection and drug discovery. The process of biomarker discovery in MS data must start with feature selection as the number of features in MS data is extremely large (e.g. thousands) while the number of samples is comparatively small. In this study, we propose the use of genetic programming (GP) for automatic feature selection and classification of MS data. This GP based approach works by using the features selected by two feature selection metrics, namely information gain (IG) and relief-f (REFS-F) in the terminal set. The feature selection performance of the proposed approach is examined and compared with IG and REFS-F alone on five MS data sets with different numbers of features and instances. Naive Bayes (NB), support vector machines (SVMs) and J48 decision trees (J48) are used in the experiments to evaluate the classification accuracy of the selected features. Meanwhile, GP is also used as a classification method in the experiments and its performance is compared with that of NB, SVMs and J48. The results show that GP as a feature selection method can select a smaller number of features with better classification performance than IG and REFS-F using NB, SVMs and J48. In addition, GP as a classification method also outperforms NB and J48 and achieves comparable or slightly better performance than SVMs on these data sets.

1 Introduction

Mass spectrometry (MS) is a tool for measuring the molecular masses of chemical compounds, and these masses are necessary to identify the species of proteins and metabolites [1]. Inside the instrument, the sample molecules are converted to ions in the ionization source and then these ions are passed to the mass analyzer to measure their mass to charge ratios (m/z). MS can work in either the full scan mode which measures the parent ions m/z (called MS spectrum, which contains m/z ratios and their corresponding intensities), or the tandem MS mode that selects and fragments the ion of interest from the parent ions list and subsequently records the m/z ratios of its drought ions (called MS/MS

L. Vanneschi, W.S. Bush, and M. Giacobini (Eds.): EvoBIO 2013, LNCS 7833, pp. 43–55, 2013.

spectrum, which contains the m/z ratios and their corresponding intensities of the fragment ions) to aid the elucidation of the structure of the molecules [1].

The MS machine is usually coupled with separation techniques, which can be either gas chromatography (GC) or liquid chromatography (LC), in the front to separate the different molecules before MS detection to reduce the complexity of the MS spectrum and increase the detection coverage. Accordingly, the data acquired with LC instrumental set up is called LC MS/MS spectrum, which contains the features of retention time, scan numbers, and m/z ratios and the corresponding intensities of the parent and fragment ions [1].

Due to the nature of the MS data, a data set typically has a very small number of instances but each instance is represented by a huge number of features (typically several thousand), which makes the classification of MS data a very challenging problem [2].

Several machine learning and optimisation techniques have been used for feature selection and classification of MS data. For example, principle component analysis and linear discriminant analysis [3, 4] and random forest algorithm [5] have been used for feature extraction, dimensionality reduction and classification on MS data. Support vector machines (SVMs), k-nearest-neighbour and quadratic discriminant analysis [6] have been used for MS data classification. Genetic algorithms and t-test [7] have also been used for feature selection and classification. SVM-RFE is used to select features by giving a weight to each feature through training a linear SVM and removing the lowest weight features. Feature subsets are passed to SVMs classifiers to classify the protoemics MS data sets [8]. However, many of these techniques for classification cannot easily handle such a huge number of features, and feature selection and dimensionality reduction of data often need to take place before classification is performed. However, the two separate processes cannot be connected so well sometimes, for example, the features selected by one method (e.g. decision trees) do not perform well for classification using a different method (e.g. SVMs).

Genetic programming (GP) is one of the evolutionary computation algorithms [9]. It starts with a random initial population to search for a solution to a given task and over a number of generations it modifies the initial solutions through a set of genetic operators guided by the fitness function [10]. GP has the capability to select features implicitly [11] by incorporating useful features during the evolution of programs.

Since very recently, there have been a small number of works only using GP for feature selection and classification of bio-data (e.g. [12,13]). For example, GP has also been used for peptide quantification of MS data and the measurement of protein quantities within a biological sample [12]. However, GP has been very seldom used for feature selection and classification of MS data.

The overall goal of this paper is to investigate a GP based approach to automatic feature selection and classification of MS data, which typically has a huge number of features and a small number of examples. To achieve feature selection, we will embed two existing feature selection metrics, information gain (IG) [14] and relief-f (REFS-F) [15,16], into the GP system. The features selected

by GP will be compared with the original features and the features selected by the above two metrics alone on five MS data sets using three common classifiers namely NB, SVMs, and J48 decision trees (J48). Specifically, we investigate the following objectives:

- what primitives and fitness function can be used to develop this GP system;
- whether embedding multiple feature selection metrics into GP can improve the feature selection process;
- whether GP as a feature selection method can automatically select a small number of features that can achieve better classification performance than using all the original features;
- whether the new set of features selected by GP outperforms the features selected by IG and REFS alone; and
- whether GP as a classification method outperform naive Bayes, SVMs, J48 classifiers.

The remainder of the paper is organised as follows. Section 2 describes the data sets and the preprocessing steps. Section 3 describes the new GP approach. Section 4 presents the results with discussions. Section 5 presents the conclusions and the future work.

2 Data Sets and Preprocessing

Five MS data sets are used in the experiments, as shown in Table 1. They are the high resolution SELDI-TOF ovarian cancer data set and the low resolution SELDI-TOF ovarian cancer data set[1] [17], the premalignant pancreatic cancer data set[2] [2], the Arcene data set [18], and the spike-in LC MS/MS data set [19] from Georgetown University[3].

Table 1. Data Sets

Name of the Data Set	No. of Features	Size of Data Set
Premalignant Pancreatic Cancer	6771	181
High Resolution Ovarian Cancer	15,000	216
Low Resolution Ovarian Cancer	15,154	253
Arcene	10,000	200
Spike-In	10,411	10

2.1 Data Preprocessing

Due to the nature (e.g. a lot of noise) and types (e.g. MS, LC MS/MS) of the MS data, different preprocessing methods are applied. We use the bioinformatics toolbox in Matlab [20] for this purpose.

Low and high resolution ovarian cancer data sets: To obtain the same m/z point at all MS spectra [17], the resampling algorithm in the toolbox is used. Then, the

[1] Available at: `http://archive.ics.uci.edu/ml/datasets/Arcene`
[2] Available at: `http://home.ccr.cancer.gov/ncifdaproteomics/ppatterns.asp`
[3] Available at: `http://omics.georgetown.edu/massprep.html`

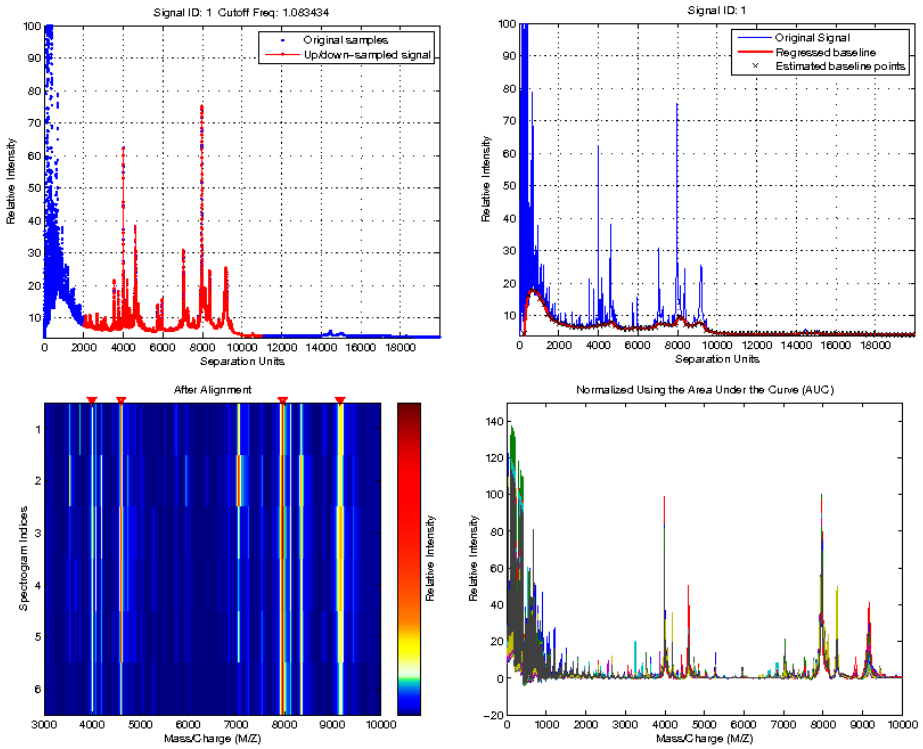


Fig. 1. Preprocessing steps of the low resolution ovarian cancer data set. (a) and (b) represent the resampling and the baseline adjustment of the first signal, respectively. (c) is an example alignment of 6 samples. Finally (d) represents the normalization of the samples using AUC.

baseline correction is adopted to subtract the background noise and remove the low intensity peaks. Firstly, the baseline is estimated by calculating of the minimum m/z values within a window size of 50 m/z points for the high resolution data. For the low resolution data, the window size is 500 m/z points. Afterwards the varying baseline is regressed and the resulting baseline is subtracted [17]. Due to mis-calibration of mass spectrometers, systematic shifts can appear in repeated experiments. Therefore, the alignment of the spectrograms is the third step in the preprocessing framework. Finally, each spectrum is normalized using area under the curve (AUC). Figure 1 shows the four steps of preprocessing of the low resolution ovarian cancer data set as an example.

Premalignant pancreatic cancer data set: Similar to the steps of the previous two data sets, baseline adjustment, filtering and normalization are used here. Baseline adjustment is estimated by segmenting the whole spectra into windows with a size of 200 m/z ratio intensities. The mean values of these windows are then used as the estimate of the baseline value at that intensity. To perform regression, a piecewise cubic interpolation method is used [2]. After this step, noise is filtered using the

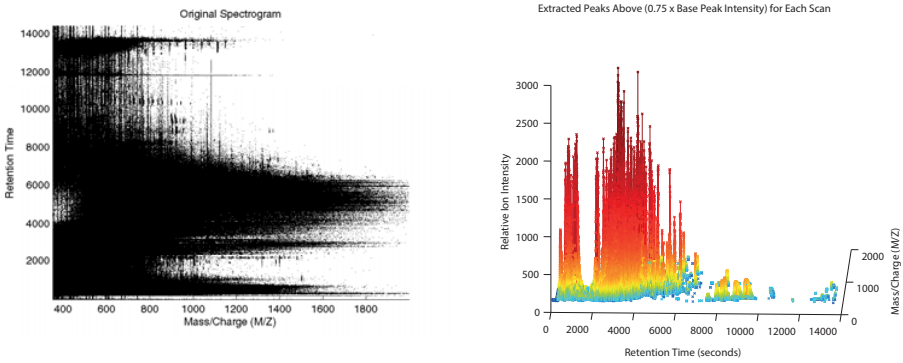


Fig. 2. Peak extraction from raw data spectrum of the spike-in data set

Gaussian kernel filter. Finally, normalization is performed using area under curve where the maximum value of intensity for each m/z ratio is rescaled to 100.

Arcene data set: This data set is available after preprocessing, where the m/z is limited between 200 and 10,000 [18] by considering the m/z values under 200 and over 10,000 as noise. Afterward, the technical repeats are averaged, baseline is removed and then smoothing and alignment take place.

The preprocessing of these four MS data sets is at a low level, therefore the number of features remains the same after preprocessing.

Spike-in data set: The data set is an LC MS/MS data set, which consists of values of scan numbers, LC retention time, m/z ratios and the corresponding intensities to the m/z ratios [1]. Ideally, the same molecule detected in the same LC MS/MS should have the same molecular weight, intensity and retention time, but due to experimental condition variation and the co-elution of molecular ions this is not always the case. Therefore, before computational analysis can take place, the data has to undergo different preprocessing steps than the steps for the full scan MS data. The first step is to extract peaks from the data by clustering significant peaks and noisy peaks and removing the noisy peaks using the toolkit. Figure 2 shows the raw data and the data after peak extraction. The second step is to filter the peaks to further remove noise from each scan. Using a percentile of the base peak intensity the filtering is performed, where the base peak is the most intense peak found in each scan. In order to produce the centroid data from the raw signal, the peak preserving resampling method is adopted. Finally the alignment of the peaks is used to remove the fluctuation of the data. After the preprocessing, the number of features becomes 847 for this data set.

3 The Approach

There are many feature selection methods used for dimensionality reduction and improving classification accuracy. Each of these methods can select different sets of features according to the criteria of the selection process. Some of the features

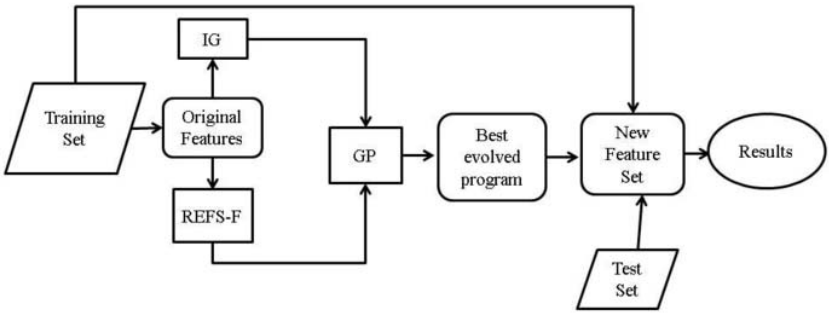


Fig. 3. Overview of the approach

selected may be powerful while other features may not be so relevant [21]. We hypothesize that combining features of different feature selection metrics may improve the feature selection performance. Our aim is to use GP to guide the feature selection process by combining two well known feature selection metrics, IG and REFS-F, and to produce a new and smaller set of features that can effectively improve the power of selected features in terms of classification accuracy. The two metrics are chosen due to their wide applications in the literature [14,15] and also because they show distinct characteristics.

The tree-based GP [11] was used here and the proposed method has three steps: (1) we use the two feature ranking techniques IG and REFS-F to rank the features; (2) the top features ranked by the two metrics are used as terminals of the GP method, where the intrinsic capability of GP is used to search for good combinations of the features from those features to form a (hopefully) better set of features; (3) the features selected by the best GP evolved program are used by the three classifiers (and GP) for classification. Figure 3 shows an overview of the proposed GP approach. In the rest of this section, we will describe the feature selection metrics, the terminal set, the function set, the fitness function, and parameter settings for the proposed method.

3.1 Feature Selection Metrics

The two feature selection metrics, IG [14] and REFS-F [15,16], are used to rank the importance of the individual features. We briefly describe them here.

Information Gain (IG) determines the amount of information gained about a class when a certain feature is present or absent [21]. It is defined as follows:

$$IG(\grave{X}, C_i) = \sum_{C \in \{C_1, C_2\}} \sum_{\grave{X} \in \{X, \bar{X}\}} P(\grave{X}, C) \frac{\log P(\grave{X}, C)}{P(\grave{X}) P(C)} \tag{1}$$

where X and $\bar{X}$ denotes the presence and the absence of the feature, and the healthy and diseased classes are denoted by $C1$ and $C2$.

Relief-F (REFS-F) searches for the two nearest neighbors for a given example, one from the same class (hit) and the other from a different class (miss) [15, 16] and calculates the worth of the feature, which is given by:

$$W(X) = \text{P(different value of X} \mid \text{nearest example from the different class)} - \text{P(different value of X} \mid \text{nearest example from the same class)} \quad (2)$$

where P refers to probability, and a good feature should differentiate between instances belonging to different classes and should have the same value for the examples from the same class.

3.2 Terminal and Function Sets

The MS data is represented by $(m/z, Int) = (m/z, Int_1, ..., Int_n)$, where m/z is a vector of the measured m/z ratios, and Int_i is the corresponding intensity for the ith sample. The objective is to predict the class label based on the intensity profile [22]. For the five data sets used, there are two classes, and the class labels can be defined as class1 or class2, respectively.

Terminal Set. As stated earlier, the goal of GP is to further select a smaller number of features from the feature pool selected by IG and REFS-F. The rationale is as follows. Firstly, the best features which are individually good are often correlated or redundant. The combination of all individually high-rank features often does not perform as well as the mixture of some individually high-rank and low rank features together. Secondly, the two metrics IG and REFS-F rank individual features based on different criteria as stated above, and accordingly we expect the combinations of the two groups of features might lead to better performance. Thirdly, using GP to further select features from the feature pool selected by IG and REFS-F can reduce the search space and possibly computational cost. Finally, GP has an implicit feature selection capability, and we expect GP to automatically select some individual features from those chosen by the two metrics and combine them together via the operators in the function set to form a small feature set that can result in better classification performance. Thus in this approach, we used the top 50 features from each of the two metrics (IG and REFS-F) to form the terminal set with 100 feature terminals, in addition to randomly generated constant terminals.

Function Set. Besides the four commonly used basic arithmetic operations $+, -, \times$ and %, we also used the square root $\sqrt{\ }$, max functions, and a conditional operator $ifte$. The % is a protected division which performs the same division operation except that the result of division by zero returns zero. The use of $\sqrt{\ }$, max and $ifte$ functions aims to evolve complex and non-linear functions for feature selection and classification. The $ifte$ returns the second argument if the first argument is less than zero or returns the third argument otherwise. The $\sqrt{\ }$ is also protected, where if the argument is negative, its absolute value is considered.

Table 2. GP parameter values

GP Parameter	Parameters Value
Initial Population	Ramped Half-and Half
Tree Depth	5-17
Generations	30
Mutation Rate	0.15
Crossover Rate	0.8
Elitisim	0.05
Population Size	1000
Selection Method	Tournament Method
Tournament Size	5

3.3 Fitness Function

As our main goal is to produce a subset of features that not only reduces the search space but also yields a better classification accuracy of the proposed GP approach, we define the fitness function to be the classification accuracy, which is evaluated after filtering the input data according to the subset of features selected by the evolved program. Thus the GP framework is to maximize the fitness, such that the generated programs with associated features lead to improved classification performance.

For a specific instance in the training set, if the program output is ≤ 0, the instance is classified as class1; otherwise as class2. This fitness function will be used in the GP system for both feature selection and classification, which is a single process in this approach.

3.4 Experimental Setup

In order to evaluate the performance of our proposed method for feature selection and classification, we conducted a number of experiments on the five different MS data sets. We used the ECJ package [23] for GP. The Weka package [24] was used for running IG and REFS-F for feature selection and running NB, SVMs, and J48 for classification.

For the GP system, the initial population was generated using the ramped half-and-half method [10], the individual program tree depth is minimum 5 and maximum 17, and the population size is 1000. Also tournament selection is used with a tournament size of 5. The standard subtree crossover and mutation [10] are used. Elitism is applied to make sure the best individual in the next generation is not worse than that in the current generation. The evolution will be terminated when either the fitness reaches 100% or the maximum number of generations (30 generations) is reached. The experiments on each data set are repeated for 50 independent runs with different random seeds. The features selected by the best run are used for evaluation. Table 2 summarizes the GP parameters used in our method.

All the data sets were divided into half for training and half for testing except for the spike-in data set in which the leave-one-out cross validation method was used as the number of examples in this data set is too small.

4 Results and Discussion

Table 3 shows the classification accuracy of the SVMs, NB and J48 on the five MS *test* data sets, using all the original features (Org), the top 50 features selected by IG, the top 50 features by selected REFS-F, and the features selected by the proposed GP method. The accuracy of GP as a classifier using the four sets of features are also included in this table for further comparison.

4.1 GP Feature Selection Performance

As can be seen from Table 3, in the five data sets, the numbers of features selected by GP are only 31, 41, 25, 47 and 29. These numbers of features are not only much smaller than the total numbers of original features (100 features) used by the GP system, but also smaller than the numbers of features selected by IG or REFS-F alone. The reason for selecting the top 50 ranked features is that the performance was degrading when less number of features were used.

We can also observe that in most cases, a classifier (SVMs, NB, J48 or GP) using the features selected by GP achieved much better classification performance than using all the original features. The only exception the case of SVMs in the Ovarian cancer low and Arcene data sets, which achieved slightly better performance by using all the original features. This is mainly because the SVMs method is supposed to (or claimed to) have feature selection ability, although it can not always select good features and achieve good performance on problems with a huge number of features (such as the premalignant pancreatic cancer data set). The NB and J48 methods, on the other hand, cannot cope with a huge number of original features to achieve good performance.

By further inspection and comparison, we can observe that in most cases, using the top 50 features selected by IG or REFS-F alone improved the classification performance compared to using all the original features, which indicates that most of the classifiers are more comfortable to deal with a relatively small number of features rather than thousands of features. However, in the Ovarian cancer low and Arcene data sets, the SVMs method showed worse classification performance using features selected by IG and REFS-F alone than using all the original features, suggesting that the "top" 50 features selected by IG or REFS-F alone are individually good but the combinations of them cannot perform well possibly because some individually bad features are not included in the top 50, but they might play some role in classification.

The results also show that using a smaller number of features selected by GP generally further improved the classification performance compared to using the 50 features selected by IG or REFS-F using all the four classifiers. This suggests that GP is able to select better combinations of features from the top 100 features selected by IG and REFS-F alone (50 each) and that combinations consist of a smaller number of features that led better classification performance. Further inspection of the selected feature sets reveals that GP actually selected some high-rank features and also some low-rank features from each of the top 50 features. This is consistent with our early hypothesis that the feature

Table 3. Experimental Results

Data set	Methods	#Features	SVMs	NB	J48	GP
Ovarian cancer high	Org	15000	91.78	82.3	70.8	93.15
	IG	50	94.52	89.04	89.04	94.52
	REFS-F	50	93.15	87.67	89.04	95.89
	GP	31	93.15	93.15	91.75	**98.63**
Ovarian cancer low	Org	15154	**100.00**	80.00	85.88	94.12
	IG	50	97.64	94.12	98.82	98.82
	REFS-F	50	96.47	91.76	96.47	97.65
	GP	41	97.64	94.12	94.12	98.82
Premalignant pancreatic	Org	6771	62.22	52.22	58.88	67.78
	IG	50	63.93	62.29	54.10	75.41
	REFS-F	50	65.57	67.23	72.13	77.05
	GP	25	68.85	70.49	63.93	**78.69**
Arcene	Org	10000	84.00	74.00	66.00	77.00
	IG	50	76.00	71.00	81.00	81.00
	REFS-F	50	75.00	73.00	73.00	84.00
	GP	47	81.00	76.00	74.00	**87.00**
Spike-in	Org	847	100.00	50.00	75.00	100.00
	IG	50	100.00	100.00	50.00	100.00
	REFS-F	50	100.00	100.00	100.00	100.00
	GP	29	100.00	100.00	100.00	100.00

subset combining individually good and not-very-good features can lead to better classification performance.

4.2 GP Classification Performance

To investigate the ability of GP for classification, in the last column in Table 3, we also included the results of GP as a classifier using the four different sets of features as terminals (all plus random constant terminals). Table 3 shows that GP as a classifier generally performed much better than NB and J48 for almost all the four feature sets. Compared to SVMs, GP still performed much better in most cases, except for the Spike-in data set, where both of them achieve the ideal performance, and for the Ovarian low resolutions data set, where SVMs performed better than GP when using all the original feature sets. These results demonstrate that GP as a classifier can perform better than the NB and J48 classifiers, and compatible with or even better than SVMs for these problems. The good performance of GP and SVMs might be because SVMs is primarily developed for binary classification, and tree based GP is also good for binary classification due to the natural splitting of the program output space between positive and negative values for the two classes. Another fact is that GP has a natural ability of constructing high-level features from the original low-level features using the operators in the function set, which might also contribute to the good performance. These will need to be further investigated in the future.

4.3 Further Discussions

In the spike-in data set, human experts identified 13 features (bio-markers) that can successfully solve the problem. The proposed GP system successfully detected 6 of the 13 features with 100% accuracy. This suggests that there exist

other good combinations of features that domain experts could not identify. In other words, the proposed GP system has the potential to guiding humans to identify biomarkers. This interesting topic will also be investigated in the future.

Inspection of the details of performance on both the training and the test sets for the five problems reveals that all the four classification methods generally have an overfitting problem: the classification accuracy on the test set was considerably worse than that on the training set. This is mainly because these bio-data sets typically have a huge number of features while only a small number of instances. Clearly, more instances are required for training the classifier to reduce overfitting. However, due to the nature of the biological experiments for generating the MS data, it is impractical to substantially increase the number of instances. Feature selection approaches can improve this situation as demonstrated in the present study. Further investigation is necessary to completely solve this problem.

5 Conclusions and Future Work

The main goal of this paper was to investigate a GP approach to automatic feature selection and classification for the MS data that is characterized with a huge number of features and a small number of instances. This goal was successfully achieved by developing a GP system that takes the top features selected by IG and REFS-F alone and random constants as terminals, and the classification accuracy as the fitness function. The results show that the proposed GP approach selected a smaller number of features than IG and REFS-F, and these selected features resulted in better classification performance than the top features selected by IG and REFS-F alone and all the original features on the five problems using the SVMs, NB, J48 and GP classifiers. GP as a classifier also generally outperformed the other three classifiers on these five data sets.

The results also suggest that combining multiple feature selection metrics using GP can improve the classification performance. As future work, we will investigate whether combining more metrics using GP can further improve the classification performance. This will relate to another interesting but challenging research direction, i.e. automatic construction of high-level features from low-level features for feature/dimension reduction. In addition, we will further investigate the feature selection ability of GP for MS data to address the overfitting problem in MS data with a small number of examples.

References

1. Listgarten, J., Emili, A.: Statistical and computational methods for comparative proteomic profiling using liquid chromatography-tandem mass spectrometry. Mol. Cell. Proteomics 4, 419–434 (2005)
2. Ge, G., Wong, G.W.: Classification of premalignant pancreatic cancer mass-spectrometry data using decision tree ensembles. BMC Bioinformatics 9(1), 275 (2008)

3. Lin, Q., Peng, Q., Yao, F., Pan, X.F., Xiong, L.W., Wang, Y., Geng, J.F., Feng, J.X., Han, B.H., Bao, G.L., Yang, Y., Wang, X., Jin, L., Guo, W., Wang, J.C.: A classification method based on principal components of seldi spectra to diagnose of lung adenocarcinoma. PLoS ONE 7, e34457 (2012)
4. He, S., Cooper, H.J., Ward, D.G., Yao, X., Heath, J.K.: Analysis of premalignant pancreatic cancer mass spectrometry data for biomarker selection using a group search optimizer. Transactions of the Institute of Measurement and Control 34, 668–676 (2011)
5. Satten, G.A., Datta, S., Moura, H., Woolfitt, A.R., da G. Carvalho, M., Carlone, G.M., De, B.K., Pavlopoulos, A., Barr, J.R.: Standardization and denoising algorithms for mass spectra to classify whole-organism bacterial specimens. Bioinformatics 20(17), 3128–3136 (2004)
6. Wagner, M., Naik, D., Pothen, A.: Protocols for disease classification from mass spectrometry data. Proteomics 3(9), 1692–1698 (2003)
7. Li, L., Tang, H., Wu, Z., Gong, J., Gruidl, M., Zou, J., Tockman, M., Clark, R.A.: Data mining techniques for cancer detection using serum proteomic profiling. Artificial Intelligence in Medicine 32(2), 71–83 (2004)
8. Jong, K., Marchiori, E., Sebag, M., Vaart, A.V.D.: Feature selection in proteomic pattern data with support vector machines (2004)
9. Langdon, W.B., Poli, R., McPhee, N.F., Koza, J.R.: Genetic Programming: An Introduction and Tutorial, with a Survey of Techniques and Applications. In: Fulcher, J., Jain, L.C. (eds.) Computational Intelligence: A Compendium. SCI, vol. 115, pp. 927–1028. Springer, Heidelberg (2008)
10. Poli, R., Langdon, W.B., McPhee, N.F.: A field guide to genetic programming. Lulu Enterprises, UK Ltd. (2008)
11. Neshatian, K., Zhang, M., Andreae, P.: Genetic Programming for Feature Ranking in Classification Problems. In: Li, X., Kirley, M., Zhang, M., Green, D., Ciesielski, V., Abbass, H.A., Michalewicz, Z., Hendtlass, T., Deb, K., Tan, K.C., Branke, J., Shi, Y. (eds.) SEAL 2008. LNCS, vol. 5361, pp. 544–554. Springer, Heidelberg (2008)
12. Paul, T.K., Iba, H.: Prediction of cancer class with majority voting genetic programming classifier using gene expression data. IEEE/ACM Transactions on Computational Biology and Bioinformatics 6, 353–367 (2009)
13. Lv, Y., Guo, Y., Sun, H., Zhang, M., Wang, J.: Feature extraction using composite individual genetic programming: An application to mass classification. Applied Mechanics and Materials 198, 468–473 (2012)
14. Sebastiani, F., Ricerche, C.N.D.: Machine learning in automated text categorization. ACM Computing Surveys 34, 1–47 (2002)
15. Sun, Y., Wu, D.: A relief based feature extraction algorithm. In: SDM, pp. 188–195 (2008)
16. Kononenko, I.: Estimating Attributes: Analysis and Extensions of RELIEF. In: Bergadano, F., De Raedt, L. (eds.) ECML 1994. LNCS, vol. 784, pp. 171–182. Springer, Heidelberg (1994)
17. Petricoin, Ardekani, A.M., Hitt, B.A., Levine, P.J., Fusaro, V.A., Steinberg, S.M., Mills, G.B., Simone, C., Fishman, D.A., Kohn, E.C., Liotta, L.A.: Use of proteomic patterns in serum to identify ovarian cancer. The Lancet 359, 572–577 (2002)
18. Guyon, I., Gunn, S.R., Ben-Hur, A., Dror, G.: Result analysis of the nips 2003 feature selection challenge. In: NIPS (2004)
19. Tuli, L., Tsai, T.H., Varghese, R., Xiao, J.F., Cheema, A., Ressom, H.: Using a spike-in experiment to evaluate analysis of LC-MS data. Proteome Science 10, 13 (2012)

20. Cai, J., Smith, D., Xia, X., Yuen, K.Y.: MBEToolbox: a Matlab toolbox for sequence data analysis in molecular biology and evolution. BMC Bioinformatics 6(1), 64 (2005)
21. Sandin, I., Andrade, G., Viegas, F., Madeira, D., da Rocha, L.C., Salles, T., Goncalves, M.A.: Aggressive and effective feature selection using genetic programming. In: IEEE Congress on Evolutionary Computation, pp. 1–8. IEEE (2012)
22. Wu, B., Abbott, T., Fishman, D., McMurray, W., Mor, G., Stone, K., Ward, D., Williams, K., Zhao, H.: Comparison of statistical methods for classification of ovarian cancer using mass spectrometry data. Bioinformatics 19(13), 1636–1643 (2003)
23. White, D.R.: Software review: the ecj toolkit. Genetic Programming and Evolvable Machines, 65–67 (2012)
24. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. SIGKDD Explorations 11(1), 10–18 (2009)

Structured Populations and the Maintenance of Sex

Peter A. Whigham[1], Grant Dick[1], Alden Wright[2], and Hamish G. Spencer[1]

[1] Otago University, Dunedin, New Zealand
peter.whigham@otago.ac.nz
[2] University of Montana, Montana, USA
alden.wright@umontana.edu

Abstract. The maintenance of sexual populations has been an ongoing issue for evolutionary biologists, largely due to the two-fold cost of sexual versus asexual reproduction. Many explanations have been proposed to explain the benefits of sex, including the role of recombination in maintaining diversity and the elimination of detrimental mutations, the advantage of sex in rapidly changing environments, and the role of spatial structure, finite population size and drift. Many computational models have been developed to explore theories relating to sexual populations; this paper examines the role of spatial structure in supporting sexual populations, based on work originally published in 2006 [1]. We highlight flaws in the original model and develop a simpler, more plausible model that demonstrates the role of mutation, local competition and dispersal in maintaining sexual populations.

Keywords: sexual selection, Muller's ratchet, spatial structure.

1 Introduction

The enigma of sex remains a central problem in biology [2]. Since a sexual individual transmits only half of their genes to its offspring, compared with an asexual organism, there is a two-fold cost associated with sex. Hence, unless a sexual individual can breed twice as often as an asexual individual, there is a transmission loss that implies a disadvantage to the sexual population [3]. However, given that the majority of eukaryotes are sexual some set of factors must contribute to overcoming the basic two-fold cost from reproduction.

Two main issues relate to the question of sexual reproduction: how did sexual reproduction come into being, and what factors support the maintenance of sexual versus asexual populations. As noted by Maynard Smith [2], the question of how sexual populations arose is difficult because we cannot rerun the process. However the mechanisms involved in the maintenance of sexual populations can be modeled and therefore studied to support theories involved in their continuation over time. Consequently, a large number of theories have been published regarding this issue, including the role of recombination in maintaining diversity [4] and eliminating detrimental mutations [5], producing robust populations

L. Vanneschi, W.S. Bush, and M. Giacobini (Eds.): EvoBIO 2013, LNCS 7833, pp. 56–67, 2013.

under environmental variation, and allowing the rapid accumulation of beneficial mutations [6]. In addition, the role of population structure and drift [7] have also been considered. Clearly a combination of factors is likely to contribute to the observed sexual maintenance in real populations [8]. Detailed reviews may be found in Butlin [9], Otto [3], as well as the historical, but still largely relevant, work of Maynard Smith [2].

The purpose of this work is to critically examine the model of Salathé et al. [1] where they consider the role of spatial structure and the relationship to Muller's ratchet, in maintaining sexual populations. This paper will demonstrate that aspects of the model are inconsistent with the assumptions of the model, and that much of the work needs to be revised. In particular, we show and correct the flaws in the model, and present a new model that removes the arbitrary parameters in the original formulation. The resulting model is more parsimonious, requires fewer parameter settings, and directly addresses a number of the assumptions that were not fully considered by Salathé et al. [1]. The results demonstrate that, for a range of mutation rates and selection coefficients, spatial structure has a significant influence on reducing the two-fold cost of sex.

The structure of the paper is as follows: § 2 presents the original Salathé model; § 3 examines the assumptions in this model and shows the basic flaws that need to be addressed; § 4 describes the revised model that corrects these issues; and § 5 presents the results comparing the two models and examines the influence of spatial structure on the maintenance of sex. Finally, § 6 draws some conclusions and suggests future work.

2 The Salathé Model

The Salathé model examines the role of local dispersal and local competition and demonstrates their efficacy in reducing the likelihood of an asexual invading a population of sexual individuals. Intuitively, since a regular spatial structure (such as a grid or ring) slows the dispersion of asexual clones, the accumulation of deleterious mutations (for some parameter settings) erodes the twofold advantage of asexuality faster than the asexuals can take over the population.

The Salathé model represented space as a toroidal grid, with one individual per grid cell. Individuals were haploid with $L = 512$ mutation loci, taking one of two possible values. In addition, one locus determined the reproduction mode. The model was initialised with a population of sexual individuals containing no mutations, and run for 500 generations to obtain a mutation-selection balance. A single asexual individual was then introduced to the population by converting one randomly selected sexual individual to an asexual individual by flipping the reproduction mode locus.

Reproduction depends on whether an individual is sexual or asexual. With probability 0.5 (representing the two-fold cost of sex) a sexual individual at location (x,y) randomly selects a mate from the adjacent eight cells defined by the Moore neighborhood surrounding the cell at (x,y). If no sexual individual is within this neighborhood no reproduction occurs. Each pair of sexually reproducing individuals produce 10 offspring, where each offspring obtains a randomly

selected parent allele at each locus. Each offspring accumulates on average U mutations per generation (with no back mutations), and an individual changes its mode of reproduction with probability P (although for the simulations presented here $P = 0$). An offspring survives with probability $(1 - s)^n$, where n is the number of mutations and s is the selection coefficient due to mutation. This represents a multiplicative fitness with no interaction between loci. Surviving offspring are dispersed randomly within the Moore neighbourhood of (x,y), including the location (x,y). Salathé et al. stated that "in the rare case when a cell receives no offspring..." the parent at location (x,y) is maintained for the next generation. This process is performed for each individual in the population. Finally, the next generation population is formed by randomly selecting an offspring at each cell (if one or more exists). The simulation is run until a reproduction mode of asexual or sexual is fixed within the population. In addition, if all asexuals are lost from the population within the first 10 generations, the simulation is restarted and not included in the final measurement. The original model did not explicitly describe how asexuals were handled, so we assumed that an asexual at a location behaved in a similar manner to the sexual individuals, however there was no need to examine the neighbourhood for a mate. Hence, asexuals produced 10 offspring that were mutated, survived with probability $(1 - s)^n$, and placed randomly within the neighbourhood of the location.

3 Criticisms of the Salathé Model

There are a number of issues with the previous model: ignoring the initial effects of drift by deleting those runs in which asexuals were lost within an arbitrary 10 generations; the arbitrary finite genome length L; the arbitrary 10 offspring generated from reproducing individuals; the copying procedure that occurs in the "rare case when a cell receives no offspring"; and the implicit local competition. Each of these issues will now be examined in more detail, prior to presenting a more parsimonious and biologically justified model that addresses these issues.

3.1 Drift

The Salathé model examines the probability of an single introduced asexual taking over the initial population of sexual individuals. However, if the asexuals are lost within the first 10 generations the results are not included in the final assessment of behaviour. This procedure ignores the role of drift as a barrier to the establishment of asexuals in the population, even though drift is often presented as a fundamental process in the fixation of mutations in a population [10]. Since the mathematical model presented in [1] did not account for drift we can only assume that this initial loss of asexuals was removed to align the results more favourably with the presented simulations. The obvious result of incorporating drift is that the probability of maintaining a sexual population is increased. The effect of incorporating drift is shown in Fig. 1 and confirms that this procedure does indeed bias the result.

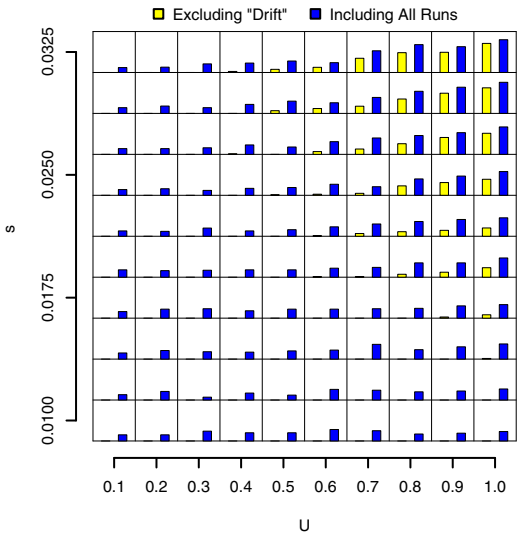


Fig. 1. The original Salathé model, using a torus and moore neighbourhood, for a range of combinations of s and U (population size $N = 400$) over 100 independent runs. The bars show the proportion of runs where sex was maintained (fixed). Note that when the early loss of asexuals due to drift is ignored the maintenance of sex is underestimated.

3.2 The Finite Genome Length

Mutation, as implemented in the Salathé model, does not allow for back mutations, nor does it allow for mutations to accumulate at a given 'site' (locus). Therefore, a new mutation must take place at a previously unmutated site which, for a finite genome of length L, will become more difficult to find as time progresses. Since the model assumes mutations occur at a random locus, some mutations will not have an effect, therefore reducing the effective mutation rate for an individual over time. Given a mutation rate of U per individual per generation, and a genome length of L, the mutation rate over time can be expressed as the combination of the probability of a mutation occurring at a locus, multiplied by the probability that a mutation has not previously occurred at that locus:

$$U'(t) = \frac{U}{L} \times \left(1 - \frac{U}{L}\right)^{(t-1)} \tag{1}$$

Therefore, the number of mutations an offspring expects to acquire in a generation is:

$$n(t) = L \times U'(t) \tag{2}$$

and the expected mutation load (under drift) is the sum of the series of mutations up to time t:

$$N(t) = \sum_{i=1}^{t} n(i) = \sum_{i=1}^{t} U \left(1 - \frac{U}{L}\right)^{(i-1)} \tag{3}$$

Equation 3 is a geometric series with solution (given $0 < \frac{U}{L} < 1$):

$$N(t) = L\left(1 - \left(1 - \frac{U}{L}\right)^t\right) \tag{4}$$

The influence of a finite genome is shown in Fig. 2 and shows that a reduction in mutation load occurs due to the finite genome length. Although the effective mutation rate reduces slowly (since $\frac{U}{L} \ll 1$) this introduces an arbitrary parameter L that could have a significant effect for long simulation runs. An alternative approach, often used in the mathematical treatments of mutation, is to assume that the number of sites in a genome is so large that each mutation occurs at a new site [11,12]. This assumption effectively means that, for the purposes of the simulation, the genome length L is infinite, which is easily accommodated in a simulation by increasing the length of a genome for an individual each time a new mutation arises, and associating a unique id for each mutation site across the entire population. Recombination now means that different site ids are combined with a probability of 0.5, representing whether the mutated site is crossed over when the offspring is produced. The main benefits of this approach are that the length of a genome does not need to be specified as part of the model, and the effective mutation rate is kept constant over the generations of the simulation.

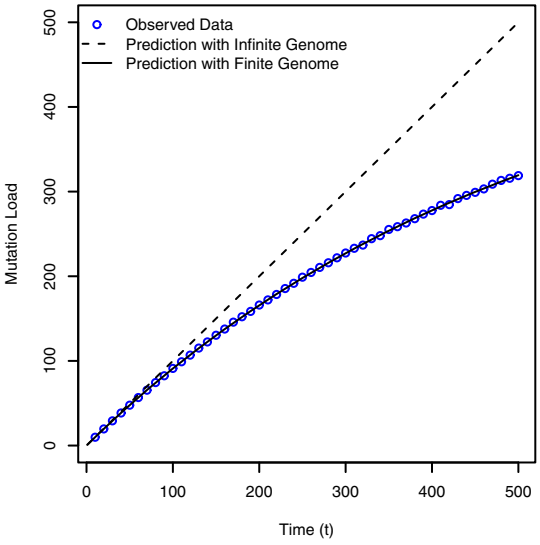


Fig. 2. The effect of a fixed-length genome ($L = 512$) on effective mutation rate under drift with $s = 0$ and $U = 1$. The observed mutation load matches well with Eqn. 4. Note the gradual reduction in mutations for the finite genome due to the loss of locii which have not been mutated.

3.3 Frequency of Rare Events

The use in the new generation of a parental type at a location where no offspring were found is, perhaps, the most significant assumption of the model. As stated in Section 2, each pair of individuals (or a single asexual) produce 10 offspring that, assuming they survive, are randomly distributed in the neighbourhood of a cell. At the completion of the reproductive phase, one offspring associated with each cell is randomly chosen to be the surviving individual for the next generation. In the *rare case* (our emphasis) that no offspring exists at a location, the previous generation individual at that location is copied. There are two serious flaws with this concept: the notion of copying an individual from one generation to the next is adhoc and biologically inappropriate (given the model is generational), and that the probability of this occurring is incorrectly assumed to be very small. This rare event will occur when:

1. The neighbouring location's occupant is sexual and chooses not to breed (taking into account the 2-fold cost of sex), and
2. Mating takes place, but a given offspring either does not survive (due to the survival probability) or is not placed at the location (individuals are randomly placed at locations within the neighbourhood).

Assume a given mean mutation load in the population of $\bar{n}$, which is equal to the sum of mutations in a population divided by the number of individuals.

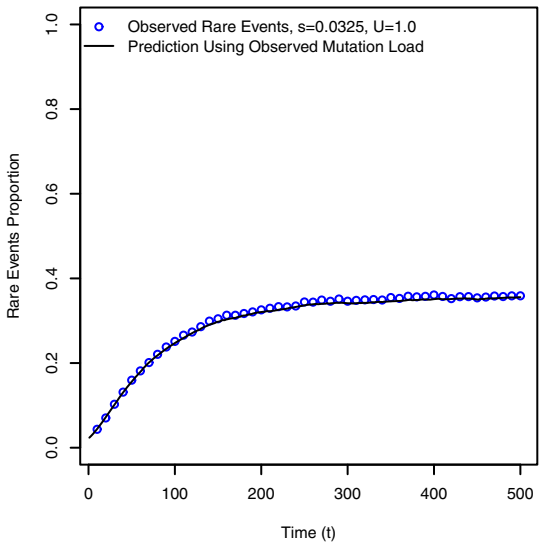


Fig. 3. The probability of a rare event occurring for the Salathé model with $s = 0.0325$ and $U = 1$. The observed data matches well with the predicted event count from Eqn. 5.

When the scenarios for a rare event are combined, the probability of rare events $p_r(\bar{n})$, given a neighbourhood of 9 cells (the current location and the immediate 8 neighbours) is given by:

$$p_r(\bar{n}) = (p_{n.p}(\bar{n}))^9 \tag{5}$$

where $p_{n.p}(\bar{n})$ is the probability of an offspring not being placed at a given location:

$$p_{n.p}(\bar{n}) = \frac{1 + (1 - p_l(\bar{n}))^{10}}{2} \tag{6}$$

This represents the situation where the probability that, of the 10 offspring that are potentially created, none land at a given location. Finally, $p_l(\bar{n})$ is the probability of placing an individual at a particular location, which is simply the chance of picking a location times the probability that it survives:

$$p_l(\bar{n}) = \frac{(1-s)^{\bar{n}}}{9} \tag{7}$$

Fig. 3 shows the count of rare events for a mutation rate $U = 1$ and selection coefficient $s = 0.0325$. The simulation shows close agreement with Eqn. 5 and demonstrates that the frequency of 'rare' events can be as high as 40% over all locations. Note that these counts are just for the first 500 generations, which is the initialisation of the model to a mutation-selection balance, and therefore it is likely that these events occur more often as additional mutations (and therefore the expected loss of individuals) increases. Although Fig. 3 shows the behaviour for a high mutation rate and selection coefficient these settings are in the range of parameter combinations where the effect of spatial structure is pronounced. Unfortunately this behaviour results in the Salathé model biasing the reproductive phase towards copying individuals from one generation to the next and therefore reduces the true influence of mutation, selection and local competition that was originally proposed.

3.4 Local Competition

The original paper proposed a model that was to investigate the role of local dispersal and local competition on the maintenance of sexual populations. Although local dispersal was explicitly represented, local competition was only modelled implicitly, and certainly did not consider direct competition between individuals. Survival was modelled using $(1-s)^n$, and this determined the offspring that survived and were placed on neighbourhood locations. However, the final determinant of which individual survived at a location in the next generation was random, assuming that more than one individual was associated with a location after all surviving offspring had been dispersed. Although this has an implicit notion of competition, since those parents with fewer mutations are likely to produce fitter children that survive and therefore are more common in the offspring pool, there is no direct competition between individuals. This seems to miss a

vital step in the evolutionary process: that survival does not necessarily lead to reproductive success, and that this is determined by the competition between individuals based on fitness. Therefore an explicit model of competition between offspring to decide which individuals are allowed to breed in the next generation should be considered.

4 A Revised Model for Sexual Populations

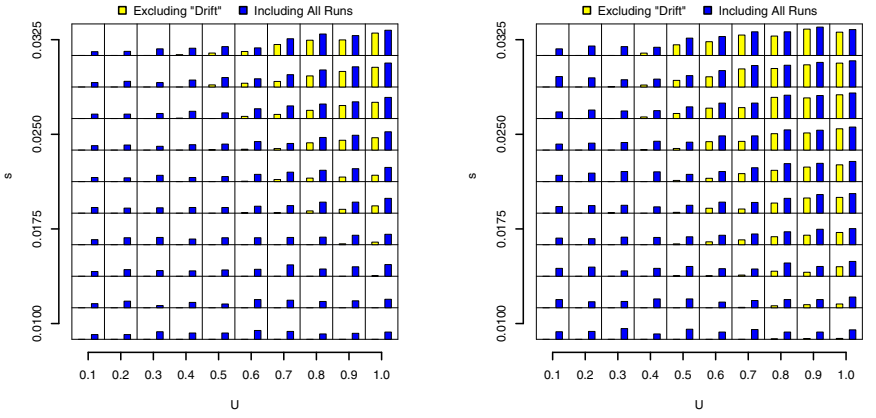
The previous section presented a number of flaws in the original formulation of the sexual model and the influence of space and recombination on Muller's ratchet. Here we present a new model that addresses the previous issues and removes a number of parameters and assumptions from the model. To make the model of space general, we consider the use of a network to represent locations, with each node representing a location, and neighbouring locations defined by the edges connecting nodes. Although a network is not necessary for a torus it allows the opportunity for the same model to be tested with a range of spatial structures [13] without having to change the underlying representation. We follow the original model in terms of a single individual at each location, a generational model, and the use of a selection coefficient s and mutation rate U. The main change to the model is with the selection of individuals and the generation of the resulting offspring at each location for the next generation. In addition, the "two-fold cost" of sexual reproduction is incorporated within the fitness model of each individual. The selection procedure to fill a particular location for the next generation, represented by a node of the network, is as follows:

1. Select one individual from the deme represented by the nodes connected to the current location (node) by an edge. Note that each location has an edge that refers to itself. The probability of selection is proportional to fitness, with fitness being $(1-s)^n$ for asexuals, and $0.5 \times (1-s)^n$ for sexual individuals, thus incorporating the two-fold cost of sex.
2. If the selected individual is asexual, then copy, mutate and place at the current location.
3. If the selected individual is sexual, then pick another sexual individual from the deme proportional to fitness (with replacement). Recombine the two selected parents, mutate the offspring and place at the current location.

This approach models selection and offspring placement by explicitly performing local competition based on fitness, and does not require the arbitrary dispersal of 10 individuals discussed in §3.3. In addition, an individual is always produced at a location, and therefore the 'rare' events problem does not occur. Note that selection for sexual individuals uses replacement and therefore allows the possibility of hermaphroditic reproduction. Since this is common in biology this operation did not seem unreasonable. In addition, allowing for hermaphroditic reproduction had no effect on the resulting survival or loss of sexuals (results not shown) and simplified the model implementation.

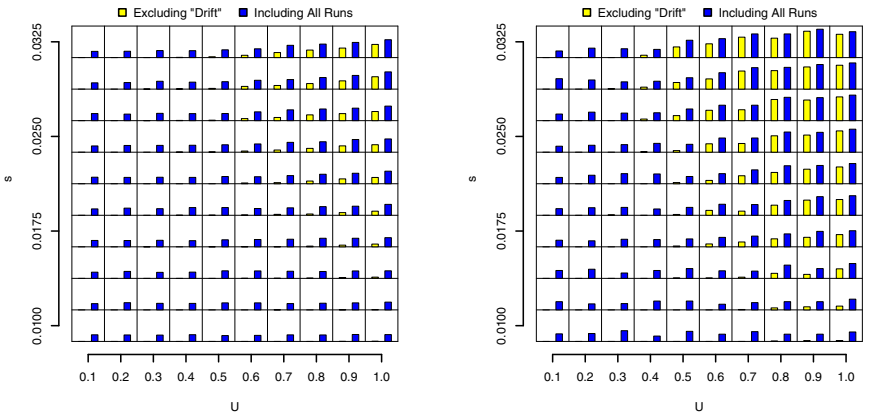
5 Results

Figure 4 shows the results of the original Salathé model and the revised model proposed in Section 4 for a 20x20 torus. These results show the revised model using an infinite genome representation (thus removing the parameter L) and with explicit local competition and deme-based dispersal. Hence there was no requirement for the arbitrary parameter defining the number of dispersed offspring (10). The revised model also addressed the issue of rare events (which have been shown to be rather common) and therefore corrected a basic flaw in the original approach. Qualitatively the models appear similar, although the revised model shows an increase in the survival of sexual populations for low values of the selection coefficient (s) and for moderate values of mutation rate (U). Given our confidence in the revised model we can now examine how a regular spatial topology affects the maintenance of sexual populations against asexual invasion. Fig. 5 shows a comparison between a well mixed (panmictic) population and a population with local dispersal on a torus. Examining the panmictic results (Fig. 5a), it is evident that with low selection pressure and low mutation rate (the lower left-hand corner), the asexuals take over the population except when they are lost through stochastic effects early in the evolution. This occurs because the two-fold cost of sex outweighs the time required for the asexuals to propagate through the population. Since each individual is part of the deme for each location, the asexuals have an early advantage in fitness. If the asexuals can survive beyond the early generations they have a significant advantage over the sexuals for low to moderate values of s and U. This result is also true for the spatial model, however the local dispersion and therefore slower rate of takeover becomes a significant disadvantage to asexuals as the number of



(a) Salathé model (b) Revised model

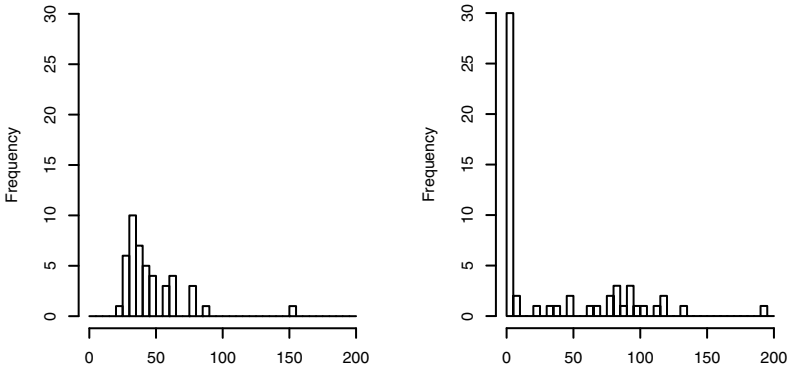
Fig. 4. Proportion of sexual populations that are maintained under a range of U and s parameters for the Salathé and revised model. For each model 100 independent runs were performed.



(a) Panmictic (b) Torus

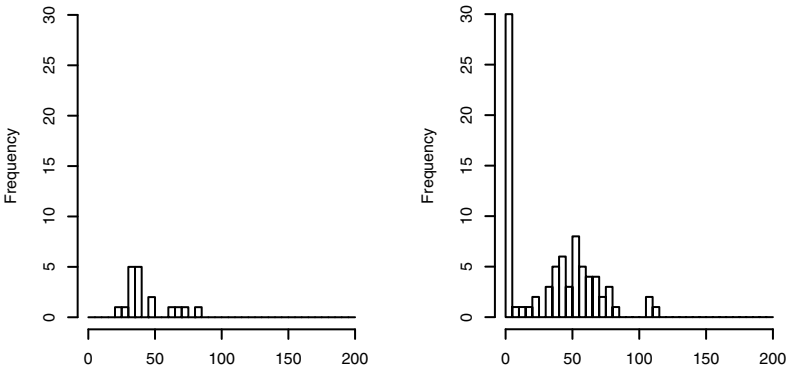
Fig. 5. Proportion of sexual populations that are maintained under a range of U and s parameters using the revised model. Results are shown for a fully-connected world (panmictic) and torus.

generations increases. For example, Fig. 6 shows the frequency and generations to fixation for asexuals and sexuals for moderate mutation rates $U = 0.5$ and $s = 0.0325$ on a torus. Note that for these settings the asexual population fixes in the population only if it occurs early in the evolution. Note that the frequency of fixation gradually reduces after ≈ 25 generations since by this stage the effect of Muller's ratchet is becoming pronounced. In contrast, the sexual population can compete with the asexuals as the evolution continues due to the reduced mutation load afforded by recombination. In addition, the large frequency count for the lowest fixation time for the sexuals indicates the loss of asexuals due to drift, clearly showing the importance of this stochastic effect as a barrier to



(a) Asexual fixation time (b) Sexual fixation time

Fig. 6. Fixation of asexuals and sexuals, $U = 0.5$, $s = 0.0325$



(a) Asexual fixation time (b) Sexual fixation time

Fig. 7. Fixation of asexuals and sexuals, $U = 1$, $s = 0.0325$

invasion. Muller's ratchet is more pronounced when the mutation rate increases, as shown in Fig. 7. Here the asexual population does not have time to propagate throughout the space before the accumulation of mutations reduces their fitness and they cannot survive in the local competition against the sexual individuals. The sexual population survives initially due to the role of drift, and subsequently due to the reduction in mutation load via recombination. The regular spatial topology slows the dispersal of individuals and therefore allows more time for the asexuals to accumulate mutations, to their obvious detriment.

6 Conclusion

This paper commenced by examining the previously published model by Salathé et al., [1] and showed the model had several basic flaws. Subsequently a revised model was presented that addressed these issues, resulting in an approach with fewer parameters and an explicit representation of local competition. In addition, the model generalised the notion of space by using a network to define the spatial topology for the local neighbourhoods. The results of the revised model demonstrated the role of drift, recombination and local dispersion in maintaining a sexual population against asexual invasion. In particular, for moderate to high mutation rates and selection coefficients, the model demonstrated that a regular spatial structure increased the probability of a sexual population surviving invasion. The generalisation of space suggests future work where a range of network structures are examined to understand how different connectivities in space relate to the fixation probability of asexuals. This can be done using a network phase-transition model that allows a broad range of spatial structures, from regular through to scale-free, to be investigated [7]. In addition, the multiplicative fitness model does not address the role of epistasis, which has previously been shown to have a role in fitness and recombination [4]. This approach, extended to our sexual maintenance model with spatial structure, would be a significant contribution to understanding why sexual populations are prevalent in biology.

References

1. Salathé, M., Salathé, R., Schmid-Hempel, P., Bonhoeffer, S.: Mutation accumulation in space and the maintenance of sexual reproduction. Ecology Letters 9, 941–946 (2006)
2. Maynard Smith, J.: The evolution of sex. Cambridge University Press (1978)
3. Otto, S.: The evolutionary enigma of sex. American Naturalist 174, S1–S14 (2009)
4. Roze, D.: Diploidy, population structure, and the evolution of recombination. American Naturalist 174, S79–S94 (2009)
5. Muller, H.: Our load of mutations. American Journal of Human Genetics 2(2), 111–176 (1950)
6. Barton, N., Charlesworth, B.: Why sex and recombination? Science 281, 1986–1989 (1998)
7. Campos, P., Cambadao, J., Dionisio, F., Gordo, I.: Muller's ratchet in random graphs and scale-free networks. Physical Review E 74, 042901-1–042901-4 (2006)
8. West, S., Lively, C., Read, A.: A pluralist approach to sex and recombination. J. Evol. Biol. 12, 1003–1012 (1999)
9. Butlin, R.: The costs and benefits of sex: new insights from old asexual lineages. Nature Reviews Genetics 3, 311–317 (2002)
10. Kimura, M.: The neutral theory of molecular evolution. Cambridge University Press (1985)
11. Kimura, M.: The number of heterozygous nucleotide sites maintained in a finite population due to steady flux of mutations. Genetics 61(4), 893–903 (1969)
12. Tajima, F.: Statistical method for testing the neutral mutation hypothesis by dna polymorphism. Genetics 123, 585–595 (1989)
13. Whigham, P., Dick, G., Spencer, H.: Genetic drift on networks: Ploidy and the time to fixation. Theoretical Population Biology 74(4), 283–290 (2008)

Hybrid Multiobjective Artificial Bee Colony with Differential Evolution Applied to Motif Finding

David L. González-Álvarez and Miguel A. Vega-Rodríguez

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

Abstract. The Multiobjective Artificial Bee Colony with Differential Evolution (MO-ABC/DE) is a new hybrid multiobjective evolutionary algorithm proposed for solving optimization problems. One important optimization problem in Bioinformatics is the Motif Discovery Problem (MDP), applied to the specific task of discovering DNA patterns (motifs) with biological significance, such as DNA-protein binding sites, replication origins or transcriptional DNA sequences. In this work, we apply the MO-ABC/DE algorithm for solving the MDP using as benchmark genomic data belonging to four organisms: drosophila melanogaster, homo sapiens, mus musculus, and saccharomyces cerevisiae. To demonstrate the good performance of our algorithm we have compared its results with those obtained by four multiobjective evolutionary algorithms, and their predictions with those made by thirteen well-known biological tools. As we will see, the proposed algorithm achieves good results from both computer science and biology point of views.

Keywords: Artificial bee colony, differential evolution, multiobjective optimization, motif discovery, DNA.

1 Introduction

In recent years, a large number of papers that propose the combination of different techniques or algorithms when designing metaheuristics is emerging. This concept is known as hybridization [10]. In many cases the combination of different techniques or algorithms allows us to cover a greater percentage of the search space defined by an optimization problem. Taking into account this possibility, we have designed a new hybrid evolutionary algorithm named Multiobjective Artificial Bee Colony with Differential Evolution (MO-ABC/DE). It combines the operation of Artificial Bee Colony (ABC, [11]) with the properties of Differential Evolution (DE, [19]). ABC is a swarm-based evolutionary algorithm motivated by the behaviour of honey bee swarms. It tries to take advantage of the collective behaviour of bees for finding good quality solutions. The MO-ABC/DE

L. Vanneschi, W.S. Bush, and M. Giacobini (Eds.): EvoBIO 2013, LNCS 7833, pp. 68–79, 2013.

algorithm follows the same scheme as ABC, but incorporating the DE operators. These operators are used to generate new candidate solutions by combining existing ones according to a simple formulation.

Both algorithms (ABC and DE) are originally designed to address problems that optimize a single objective function. However, the vast majority of real optimization problems needs to optimize simultaneously two or more conflicting objective functions subject to certain constraints. These problems are known as Multiobjective Optimization Problems (MOP) [3]. For this reason, we have also adapted our algorithm (MO-ABC/DE) to the multiobjective context, allowing it to solve real problems which optimize more than one objective function. To demonstrate the effectiveness of the MO-ABC/DE algorithm, we have tried to solve an important biological problem, the Motif Discovery Problem (MDP). This problem has been formulated as a multiobjective optimization problem and aims to find DNA patterns - motifs - that have some biological significance, such as DNA-protein binding sites, replication origins or transcriptional DNA sequences [5]. To optimize it, we must maximize three conflicting objectives: the motif length, support and similarity; while fulfilling some constraints. The results obtained by our hybrid algorithm are evaluated and analysed in a broad comparative section. First we compare the results obtained by the MO-ABC/DE algorithm with those achieved by the Multiobjective Artificial Bee Colony (MOABC, [9]) and the Differential Evolution with Pareto Tournaments (DEPT, [8]). Thus we demonstrate the advantages of the hybridization. And then, we compare the MO-ABC/DE with the Non-dominated Sorting Genetic Algorithm II (NSGA-II, [4]) and the Strength Pareto Evolutionary Algorithm 2 (SPEA2, [22]) algorithms. Finally, we also compare the predictions made by our algorithm with those predicted by thirteen well-known biological tools. As we will see, MO-ABC/DE achieves good results from both computer science and biology point of views.

The rest of the paper is organized as follows. In the following section we define the MDP, including a brief review of the problem state-of-the-art. In Section 3 we detail the proposed algorithm, explaining its operation and the adjustments made for its multiobjective adaptation. The experimental methodology and results are included in Section 4. In this section we also compare our algorithm with other multiobjective evolutionary algorithms. Then, in Section 5 we compare the predictions made by the MO-ABC/DE algorithm with those predicted by other thirteen biological tools. Finally, some conclusions and future lines are discussed in Section 6.

2 Motif Discovery Problem

In this section we include a review of works related to the resolution of the MDP. We also describe the MDP mathematical formulation, explaining the defined objectives and constraints.

2.1 MDP State-of-the-Art Review

The MDP is extensively addressed in the literature. Among the different existing biological tools we can distinguish the string-based tools. These tools are appropriate for finding totally constrained motifs, some examples are Oligo/Dyad-Analysis, MITRA (Mismatch Tree Algorithm), YMF, QuickScore, or Weeder. Other tools are based on probabilities. These methods are usually designed to find longer or more general motifs. The most popular methods are Consensus, MEME, AlignACE, ANN_Spec, Improbizer, MotifSampler, GLAM, or SeSiMCMC. Thanks to the work [20], we can compare the results obtained by these biological tools.

In recent years, several evolutionary algorithms to discover motifs are emerging. In [13] the authors propose a genetic algorithm (GA) for finding motifs, the FMGA. Another GA-based method is St-GA (Structured Genetic Algorithm) proposed in [18]. [1] also proposes a GA called MDGA to predict binding sites. In [2] and [15] the authors developed another two GA-based approaches to find motifs. Although there are other proposals such as TS-BFO [17] which integrates Bacterial Foraging Optimization (BFO) and Tabu Search (TS) or the population clustering evolutionary algorithm (PCEA) proposed in [14]; almost all the mentioned evolutionary algorithm are based on genetic algorithms. Furthermore, all of them consider a single objective and the motif length is given beforehand, assuming only one motif per sequence. Moreover, almost all of the algorithms try to find motifs in all the given sequences. To our knowledge, the best way to address some of the problems previously listed is using a multiobjective optimization. The author of [12] proposed a multiobjective GA based method named MOGAMOD for discovering motifs, demonstrating the advantages of multiobjective optimization to discover motifs. Due to the advantages of the multiobjective optimization, we also use it in order to solve the previously cited problems.

2.2 Mathematical Formulation

Given a set of sequences $S = \{S_i | i = 1, 2, ..., D\}$ of nucleotides defined on the alphabet $B = \{A, C, G, T\}$. $S_i = \{S_i^j | j = 1, 2, ..., w_i\}$ is a sequence of nucleotides, where w_i is its sequence width. The set of all the subsequences contained in S is $\{s_i^{j_i} | i = 1, 2, ..., D, j_i = 1, 2, ..., w_i - l + 1\}$, 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. We refer to the number of motif instances as $|A| = \sum_{i=1}^{D} \sum_{j=1}^{w_i} A_i^j$. To obtain the objective values we have to build the consensus motif, which is a string abstraction of the motif instances. To compose this motif we have to take into account the dominant bases (A, C, G, or T) of the motif instance nucleotides. For example, considering a final motif composed by five motif instances with the following bases in their first positions: {A, A, C, A, G, A}, the first nucleotide of the consensus motif is "A", as it is the dominant nucleotide. This process is repeated for all positions, and in case of tie, we choose one of the dominant bases randomly. $S(A) = \{S(A)_1, S(A)_2, ..., S(A)_{|A|}\}$ is a set of $|A|$ motif instances, where $S(A)_i = S(A)_i^1 S(A)_i^2 ... S(A)_i^l$ is the ith motif

instance in $|A|$. $S(A)$ can also be expanded as $(S(A)^1, S(A)^2, ..., S(A)^l)$, where $S(A)^j = S(A)_i^j \ S(A)_2^j \ ... \ S(A)_{|A|}^j$ is the list of nucleotides on the jth position in the motif instances. To obtain the objective values we have also to build the Position Count Matrix (PCM) $N(A)$ with the numbers of different nucleotide bases on each position of the candidate motifs (A) which have passed the threshold marked by the support. $N(A) = \{N(A)^1, N(A)^2, ..., N(A)^l\}$, and $N(A)^j = \{N(A)_b^j | b \in B\}$, where $N(A)_b^j = |\{S(A)_i^j | S(A)_i^j = b\}|$; and the Position Frequency Matrix (PFM) $\widehat{N} = \frac{N(A)}{|A|}$, where the dominant nucleotides of each position are normalized.

Given these considerations, we have formulated the MDP as a multiobjective optimization problem where we maximize the following objectives:

Obj1) The *motif length* maximizes the number of nucleotides that compose the motifs.

Obj2) The *support* maximizes the number of sequences used to compose the final motif. Only those sequences that achieve a motif instance of certain quality, with respect to the consensus motif, were taken into account in this objective, and so, when we compose the final motif.

Obj3) The *similarity* maximizes the similarity among the subsequences that compose the final solution. To calculate its value we have to average all dominance 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 dominance value of the dominant nucleotide in column i.

Subject to certain constraints. Motifs are usually very short [5]. For this reason, we have restricted the motif length to the range [7,64]. We have also set a minimum support value of 2 for the motifs of the instances composed by 4 sequences, and of 3 for the other ones to avoid solutions with no biological relevance. Finally, we apply the complexity concept [7] expanded with the improvements suggested in [6]. 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 $i \in \{A, C, G, T\}$. For example, $n_A = 1$, $n_T = 2$, $n_G = 1$, and $n_C = 0$ can correspond to the 'ATTG' sequence, to the 'TGTA' sequence, or to all the other possible permutations. In the worst case, a candidate motif has only one type of nucleotides, and then its complexity is 0.

2.3 Real MDP Example

In this section we solve a MDP to clarify the calculation of each objective function. Given the candidate motifs presented in Figure 1.a, we can calculate the value of the first objective, *motif length = 20*. To obtain the value of the other objective functions, we have to build the consensus motif taking into account the bases of the nucleotides of each candidate motif. In this example, the consensus motif is: **CCCCTTCCCCATACTGCCCC**. Then, we check if the candidate motifs share at least a 50% of the bases with the consensus motif. In this example the ten candidates exceed this threshold, so we have a *support = 10*. Finally, to calculate the similarity we have to build the PCM (Figure 1.b) and the PFM by counting the bases of the candidate motifs that have exceeded the threshold value of support, and then apply Equation 1. The obtained value is *similarity = 71.50%*. Finally, we only have to compose the final motif (represented in Figure 1.c).

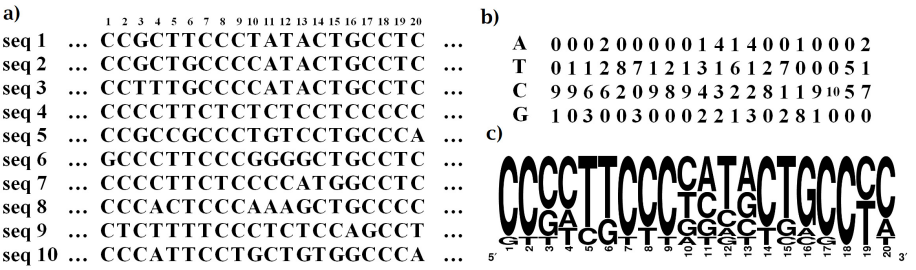


Fig. 1. An MDP example. (a) Ten possible candidate motifs. (b) Position Count Matrix. (c) Sequence logo showing the scaled frequencies relative to the information content at each position.

3 Multiobjective Artificial Bee Colony with Differential Evolution

The Multiobjective Artificial Bee Colony with Differential Evolution (MO-ABC/DE) algorithm is a new hybrid metaheuristic that combines the operation of Artificial Bee Colony (ABC - [11]) and Differential Evolution (DE - [19]).

The ABC algorithm is a swarm-based algorithm inspired by the behaviour of honey bees. ABC defines three kind of bees within the colony (algorithm population). Employed bees exploit food sources (solutions to the optimization problem), and then report on the colony the quality of the flowers found. Meanwhile, onlooker bees analyse the information provided by employees and explore around the best solutions. Finally, scout bees examine more remote areas of the colony. ABC defines two important parameters: colony size (NP) and *limit*. The latter indicates when a food source is exhausted and thus it stops exploiting the current solution. On the other hand, the DE algorithm addresses an optimization problem by iteratively trying to improve a candidate solution with

respect to a quality measure. For doing this, it takes advantage of the differences among the individuals of the population. This algorithm bases its behaviour on a set of simple crossover-mutation schemes that have been successfully applied in many research fields. These schemes have been integrated into our algorithm. DE defines three important parameters: the crossover factor (CR), the mutation factor (F), and the selection crossover-mutation scheme. To sum up, the main objective of the MO-ABC/DE algorithm is to combine the general schema of ABC with the crossover-mutation DE schemes that such successful results have provided.

3.1 MO-ABC/DE Features

The operation of the MO-ABC/DE algorithm is shown in Figure 2. As we previously explained, this algorithm define three kind of bees: employed, onlooker, and scout bees. MO-ABC/DE first initializes the food sources (solutions) that will be exploited by the employed bees (first half of the colony). Then, the algorithm tries to improve them by applying the crossover-mutation operators of DE. If the generated solutions are better than existing ones, we exchange them. After doing this, we generate a probability vector that allows us to choose employed food sources according to their qualities, so that it is more likely to select a good solution. This probability vector is used to assign the food sources to the onlooker bees (second half of the colony). These bees analyse the neighbourhood of the selected solution and select a new food source close to the selected one. To do this, we also use the crossover-mutation DE operators, but in this case, if the generated solution presents the same quality as the existing one, we also

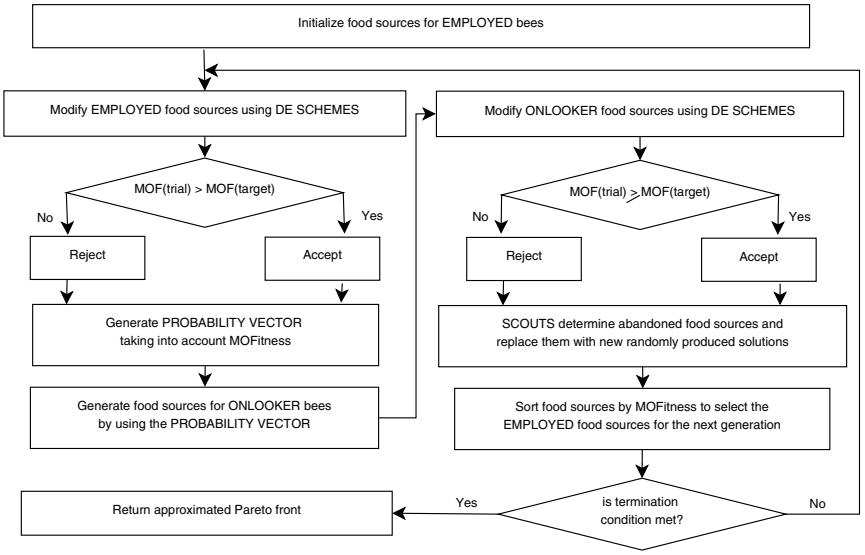


Fig. 2. Schematic flow diagram of the MO-ABC/DE algorithm

exchange them. After that, the algorithm checks which solutions do not improve (stagnated solutions) and it sends scout bees to find new solutions that will replace them. Finally, the best food sources will be kept and exploited by the employed bees in the next generation. A detailed description of its operation can be find in [16], [8], and [9].

3.2 Multiobjective Implementation Details

As we mentioned above, the ABC and DE algorithms are originally designed to address problems that optimize a single objective function. We also noted how the problem tackled in this work (MDP) is formulated as a multiobjective optimization problem. Therefore, we have adapted the MO-ABC/DE algorithm to this context, considering features and concepts used in the literature such as the dominance concept [3] or the crowding distance [4]. In addition, we have also designed a new metric to assess multiobjective solutions that has to be maximum:

$$MOF(x_i) = (2^{x_i.rank} + \frac{1}{1 + x_i.cw})^{-1} \quad (3)$$

To calculate the MOF of a given solution x_i we need to know the Pareto front to which it belongs (for example, if it belongs to the first Pareto front, it has $rank = 1$). Then, we also have to calculate its crowding distance with respect to the solutions of the same Pareto front, i.e. all solutions with $rank = 1$. Applying this expression, we can know if a solution is better than others regardless of the Pareto front to which they belong. It is applied in the DE crossover-mutation operators, where we need to know if the new generated solution is better than the existing one. We also use it for sorting the solutions at the end of each generation. Note that, the MO-ABC/DE algorithm also defines an archive file for storing the non-dominated solutions discovered in an execution.

4 Experimental Results

In this section we detail the methodology followed in our experimentation and analyse the first obtained results. To do this, we compare the results obtained by the MO-ABC/DE algorithm with those obtained by other four multiobjective evolutionary algorithms: Multiobjective Artificial Bee Colony (MOABC, [9]), Differential Evolution with Pareto Tournaments (DEPT, [8]), Non-dominated Sorting Genetic Algorithm II (NSGA-II, [4]), and Strength Pareto Evolutionary Algorithm 2 (SPEA2, [22]). The parameters of each algorithm have been properly configured after numerous experiments. In our algorithm we consider a colony size (NP) of 50 individuals, a crossover probability (CR) of 25%, a mutation factor (F) of 3%, and $limit$ equal to 30 generations. The parameter settings of the other algorithms are the same as those considered in [8] and [9]. For all experiments we have carried out 30 independent runs using g++ (GCC) 4.4.5 on a 2.3GHz Intel PC with 1GB RAM. As quality metrics we have used the

Table 1. Data set properties

	#Seq.	Size	#Nucl.	Time (s)
dm01g	4	1500	6000	20
dm04g	4	2000	8000	20
dm05g	5	2500	12500	20
hm03r	10	1500	15000	30
hm04m	13	2000	26000	30
hm16g	7	3000	21000	20
mus02r	9	1000	9000	20
mus03g	4	1500	6000	20
mus07g	12	500	6000	30
yst03m	8	500	4000	20
yst04r	7	1000	7000	20
yst08r	11	1000	11000	30

	Seq. 1	Seq. 2	Seq. 3		Seq. n
Motif Length	S_1	S_2	S_3	...	S_n

Fig. 3. Individual representation

hypervolume [23], using as benchmark the theoretical optimum of each instance; and the set coverage metric [21]. The representation of the individuals includes the necessary information to form a possible motif (Figure 3). It is represented by the motif length and the starting locations of each candidate motif in each sequence. Finally, we have used a set of twelve biological instances as benchmark with genomic data from four organisms: drosophila melanogaster (dm), homo sapiens (hm), mus musculus (mus), and saccharomyces cerevisiae (yst) obtained from [20]. Their properties are detailed in Table 1.

The hypervolumes obtained by the multiobjective evolutionary algorithms are shown in Table 2. In this table we include the median hypervolumes and the interquartile ranges of the 30 executions carried out with the configured algorithms. If we analyse the obtained results we can note how MO-ABC/DE achieves higher hypervolumes than the other four algorithms in nine of the twelve solved instances. In addition, our algorithm achieves better results than MOABC and DEPT in ten of the twelve instances, demonstrating the advantages of our proposed hybrid. Finally, we can see how our algorithm achieves an average hypervolume 5% greater than the other algorithms. To demonstrate that the differences among the results obtained by the algorithms are statistically

Table 2. Median hypervolumes and Interquartile Range (IQR) achieved by the algorithms

	MO-ABC/DE	MOABC	DEPT	NSGA-II	SPEA2
Instances	$\tilde{HV}_{IQR}$	$\tilde{HV}_{IQR}$	$\tilde{HV}_{IQR}$	$\tilde{HV}_{IQR}$	$\tilde{HV}_{IQR}$
dm01g	**89.32% 0.006**	83.24% 0.007	79.68% 0.017	81.56% 0.007	83.17% 0.007
dm04g	**86.15% 0.007**	84.14% 0.008	79.74% 0.018	81.06% 0.012	82.67% 0.011
dm05g	**89.77% 0.003**	86.43% 0.008	81.95% 0.014	84.41% 0.009	86.13% 0.011
hm03r	**73.95% 0.049**	61.48% 0.013	65.33% 0.030	47.40% 0.040	53.22% 0.016
hm04m	58.16% 0.017	56.50% 0.017	**61.25% 0.030**	43.32% 0.034	46.59% 0.009
hm16g	79.73% 0.010	**81.91% 0.032**	79.72% 0.044	68.12% 0.010	72.40% 0.017
mus02r	**70.93% 0.024**	64.17% 0.018	69.96% 0.019	59.24% 0.012	59.68% 0.015
mus03g	**81.06% 0.003**	79.69% 0.006	77.49% 0.016	77.18% 0.005	77.69% 0.005
mus07g	88.59% 0.010	88.29% 0.023	80.58% 0.034	87.01% 0.019	**89.50% 0.005**
yst03m	**74.93% 0.008**	69.73% 0.016	73.22% 0.012	65.52% 0.021	66.45% 0.011
yst04r	**80.90% 0.004**	75.57% 0.010	74.32% 0.020	74.80% 0.005	71.72% 0.006
yst08r	**78.07% 0.015**	61.81% 0.016	68.03% 0.048	64.87% 0.017	57.22% 0.011
Average HV	79.30%	74.41%	74.27%	69.54%	70.54%

Table 3. Differences that are not statistically significant

	MOABC	DEPT	NSGA-II	SPEA2
MO-ABC/DE	-	hm16(.760)	mus07(.341)	-
MOABC		-	-	dm05(.089) mus07(.118)
DEPT			mus03(.599) mus3(.969)	-
NSGA-II				dm04(.144) mus03(.462)

Table 4. Direct comparison of the outcomes achieved by the different algorithms. Each cell gives the fraction of non-dominated solutions obtained by A covered by the non-dominated points from B.

A	DEPT	MOABC	NSGAII	SPEA2	MO-ABC/DE			
B	MO-ABC/DE				DEPT	MOABC	NSGAII	SPEA2
dm01g	47.73%	25.64%	70.73%	63.41%	56.10%	82.93%	39.47%	51.28%
dm04g	33.90%	13.04%	70.73%	60.98%	78.05%	95.12%	29.27%	44.68%
dm05g	74.19%	13.33%	80.00%	84.00%	40.00%	92.00%	39.13%	26.67%
hm03r	5.26%	3.70%	5.94%	0.00%	73.27%	77.23%	95.00%	100.00%
hm04m	1.49%	9.80%	0.00%	0.00%	66.25%	56.25%	100.00%	100.00%
hm16g	0.00%	0.00%	15.22%	10.87%	78.26%	89.13%	89.19%	87.50%
mus02r	0.81%	1.15%	13.21%	7.55%	74.53%	77.36%	89.33%	89.86%
mus03g	17.57%	16.88%	49.28%	40.58%	89.86%	97.10%	74.58%	70.49%
mus07g	2.17%	0.00%	69.44%	75.00%	88.89%	100.00%	37.04%	27.27%
yst03m	4.48%	1.94%	9.09%	20.91%	79.09%	84.55%	93.06%	89.25%
yst04r	0.00%	14.75%	20.29%	5.80%	57.97%	56.52%	72.58%	94.83%
yst08r	0.00%	6.78%	0.00%	0.00%	52.31%	46.15%	100.00%	100.00%
	15.63%	8.92%	33.66%	30.76%	69.55%	79.53%	71.55%	73.49%

significant, we have studied the results included in Table 2 by pairs. In this study we apply a set of tests to analyse the distribution of the samples (Kolmogorov-Smirnov) and the homogeneity of variances (Levene). If both tests are positive we apply the ANOVA parametric test; on the other hand, if any test is negative, we apply the Kruskal-Wallis non-parametric test, always considering a confidence level of 95%. The results obtained in this statistical study are shown in Table 3. Finally, we analyse the results by using a second multiobjective metric, the set coverage. This metric allows us to know the quality of the solutions obtained by each algorithm. Given two Pareto fronts of two algorithms (A and B), the set coverage calculates the percentage of solutions of A that are able to cover any solution of B. We say that a solution x covers another solution y, if and only if x dominates y or both solutions belong to the same Pareto front. The results are included in Table 4, and they show how the MO-ABC/DE algorithm achieves higher mean coverages than the other algorithms.

5 Comparison with Other Biological Tools

In this section we compare the solutions discovered by our algorithm with the predictions made by thirteen well-known biological tools such as Consensus, MEME, AlignACE, ANN_Spec, Improbizer, MotifSampler, GLAM, SeSiMCMC, Oligo/Dyad-Analysis, MITRA, YMF, QuickScore, and Weeder. For doing this,

Table 5. Comparison between the solutions of our algorithm and the results predicted by thirteen well-known biological tools ("-" when no tool is able to find solutions)

Sensitivity (nSn)			
Instances	Best tool	Result	MO-ABC/DE
dm01g	SeSiMCMC	0.344000	**0.496000**
dm04g	MotifSampler	0.022222	**0.325926**
dm05g	MEME	0.037500	**0.193750**
hm03r	MEME	0.063726	**0.323529**
hm04m	AlignACE	0.005952	**0.238095**
hm16g	-	0.000000	**0.439024**
mus02r	MEME	0.094828	**0.387931**
mus03g	AlignACE	0.281690	**0.676056**
mus07g	ANN_Spec	0.040000	**0.630000**
yst03m	Improbizer	0.340136	**0.346939**
yst04r	Consensus	0.335878	**0.496183**
yst08r	AlignACE	**0.387097**	0.318996

Positive Predictive Value (nPPV)			
Instances	Best tool	Result	MO-ABC/DE
dm01g	SeSiMCMC	0.344000	**0.765432**
dm04g	MotifSampler	0.032967	**0.814815**
dm05g	MEME	0.026667	**0.968750**
hm03r	MEME	0.108333	**0.868421**
hm04m	AlignACE	0.006061	**1.000000**
hm16g	-	0.000000	**0.666667**
mus02r	MEME	0.142857	**0.608108**
mus03g	AlignACE	0.256410	**0.600000**
mus07g	ANN_Spec	0.020942	**0.807692**
yst03m	YMF	**0.700000**	0.607143
yst04r	MITRA	0.357143	**0.747126**
yst08r	MotifSampler	**0.786408**	0.481081

Performance Coefficient (nPC)			
Instances	Best tool	Result	MO-ABC/DE
dm01g	SeSiMCMC	0.207730	**0.430556**
dm04g	MotifSampler	0.013453	**0.303448**
dm05g	MEME	0.015831	**0.192547**
hm03r	MEME	0.041801	**0.308411**
hm04m	AlignACE	0.003012	**0.238095**
hm16g	-	0.000000	**0.360000**
mus02r	MEME	0.060440	**0.310345**
mus03g	AlignACE	0.155039	**0.466019**
mus07g	ANN_Spec	0.013937	**0.547826**
yst03m	oligodyad	0.261905	**0.283333**
yst04r	Consensus	0.202765	**0.424837**
yst08r	MotifSampler	**0.269103**	0.237333

Correlation Coefficient (nCC))			
Instances	Best tool	Result	MO-ABC/DE
dm01g	SeSiMCMC	0.330043	**0.609864**
dm04g	MotifSampler	0.013401	**0.510701**
dm05g	MEME	0.006491	**0.429207**
hm03r	MEME	0.063601	**0.523241**
hm04m	AlignACE	-0.000400	**0.486746**
hm16g	MEME	-0.005204	**0.538150**
mus02r	MEME	0.097480	**0.475147**
mus03g	AlignACE	0.222480	**0.613671**
mus07g	ANN_Spec	0.006056	**0.709135**
yst03m	oligodyad	0.437304	**0.444011**
yst04r	Consensus	0.322430	**0.603003**
yst08r	MotifSampler	**0.470596**	0.379089

we use four nucleotide-level biological metrics: the Sensitivity (nSn), the Positive Predictive Value ($nPPV$), the Performance (nPC) and Correlation (nCC) coefficients. These biological metrics are calculated by comparing the positions of the predictions made with the real binding sites in each instance through the values of TP (True-Positives), TN (True-Negatives), FP (False-Positives), and FN (False-Negatives). This comparison is made possible thanks to the work [20], where the authors provide the biological information related to the thirteen above mentioned biological tools. For further information about these tools, the used biological metrics, or the instances described in Table 1 see [20].

In Table 5 we include the results of this comparison. As we are comparing the results obtained by our algorithm with those achieved by thirteen tools by using four biological indicators, we have to process a huge amount of data. To clarify the comparisons we show the best biological tool, the results obtained by this tool, and the results achieved by our algorithm (among all non-dominated solutions, we selected the one with the best combined result); for each instance and each indicator. The values of these indicators are in the range [-1,1], where -1 indicates perfect anti-correlation and 1 indicates perfect correlation. Analysing the results of MO-ABC/DE we see how it achieves better results than the best biological tool in almost all cases. Being also important to note that while many biological tools

are specialized in certain organism instances, our algorithm is able to obtain good results in all instances, regardless of the organism to which they belong.

6 Conclusions and Future Lines

In this work we propose the use of a hybrid multiobjective evolutionary algorithm called Multiobjective Artificial Bee Colony with Differential Evolution (MO-ABC/DE) to solve an important biological problem, the Motif Discovery Problem (MDP). The proposed multiobjective algorithm combines the operations of two known algorithms, Artificial Bee Colony (ABC) and Differential Evolution (DE). To demonstrate the effectiveness of our proposed algorithm we compare its results with those achieved by other multiobjective evolutionary algorithms such as MOABC, DEPT, NSGA-II and SPEA2. In addition, we also compare the solutions discovered by MO-ABC/DE with the predictions made by thirteen well-known biological tools. Analysing the obtained results we can conclude that our hybrid algorithm improves the quality of results obtained by the four multiobjective evolutionary algorithms while making biologically relevant predictions.

As future work we will intend to apply our algorithm to solve more complex instances by using, if necessary, parallelism techniques.

Acknowledgements. This work was partially funded by the Spanish Ministry of Economy and Competitiveness and the ERDF (European Regional Development Fund), under the contract TIN2012-30685 (BIO project). Thanks also to the Fundación Valhondo for the economic support offered to David L. González-Álvarez.

References

1. 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)
2. Congdon, C.B., Fizer, C.W., Smith, N.W., Gaskins, H.R., Aman, J., Nava, G.M., Mattingly, C.: Preliminary results for GAMI: A genetic algorithms approach to motif inference. In: Proceedings of the 2005 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB 2005), pp. 97–104 (2005)
3. Deb, K.: Multi-objective optimization using evolutionary algorithms. John Wiley & Sons (2001)
4. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
5. D'haeseleer, P.: What are DNA sequence motifs? Nature Biotechnology 24(4), 423–425 (2006)
6. Fogel, G.B., et al.: Evolutionary computation for discovery of composite transcription factor binding sites. Nucleic Acids Research 36(21), 1–14 (2008)

7. Fogel, G.B., Weekes, D.G., Varga, G., Dow, E.R., Harlow, H.B., Onyia, J.E., Su, C.: Discovery of sequence motifs related to coexpression of genes using evolutionary computation. Nucleic Acids Research 32(13), 3826–3835 (2004)
8. González-Álvarez, D.L., Vega-Rodríguez, M.A., Gómez-Pulido, J.A., Sánchez-Pérez, J.M.: Predicting DNA motifs by using evolutionary multiobjective optimization. IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews 42(6), 913–925 (2011)
9. González-Álvarez, D.L., Vega-Rodríguez, M.A., Gómez-Pulido, J.A., Sánchez-Pérez, J.M.: Comparing multiobjective swarm intelligence metaheuristics for DNA motif discovery. Engineering Applications of Artificial Intelligence 26(1), 314–326 (2012)
10. Grosan, C., Abraham, A.: Hybrid Evolutionary Algorithms: Methodologies, Architectures, and Reviews. In: Grosan, C., Abraham, A., Ishibuchi, H. (eds.) Hybrid Evolutionary Algorithms. SCI, vol. 75, pp. 1–17. Springer, Heidelberg (2007)
11. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Technical report-tr06, Erciyes University, Turkey (2005)
12. Kaya, M.: MOGAMOD: Multi-objective genetic algorithm for motif discovery. Expert Systems with Applications 36(2), 1039–1047 (2009)
13. 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)
14. Lones, M.A., Tyrrell, A.M.: Regulatory motif discovery using a population clustering evolutionary algorithm. IEEE/ACM Transactions on Computational Biology and Bioinformatics 4(3), 403–414 (2007)
15. Paul, T.K., Iba, H.: Identification of weak motifs in multiple biological sequences using genetic algorithm. In: Proceedings of the 2006 Conference on Genetic and Evolutionary Computation (GECCO 2006), pp. 271–278 (2006)
16. Rubio-Largo, A., González-Álvarez, D.L., Vega-Rodríguez, M.A., Gómez-Pulido, J.A., Sánchez-Pérez, J.M.: MO-ABC/DE - multiobjective artificial bee colony with differential evolution for unconstrained multiobjective optimization. In: 13th IEEE International Symposium on Computational Intelligence and Informatics, pp. 157–162 (2012)
17. 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)
18. 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)
19. Storn, R., Price, K.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11(4), 341–359 (1997)
20. Tompa, M., et al.: Assessing computational tools for the discovery of transcription factor binding sites. Nature Biotechnology 23(1), 137–144 (2005)
21. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: empirical results. Evolutionary Computation 8(2), 173–195 (2000)
22. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength pareto evolutionary algorithm. Technical report tik-report 103, Swiss Federal Institute of Technology, Zurich, Switzerland (2001)
23. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. IEEE Transactions on Evolutionary Computation 3(4), 257–271 (1999)

ACO-Based Bayesian Network Ensembles for the Hierarchical Classification of Ageing-Related Proteins

Khalid M. Salama and Alex A. Freitas

School of Computing, University of Kent, UK
{kms39,A.A.Freitas}@kent.ac.uk

Abstract. The task of predicting protein functions using computational techniques is a major research area in the field of bioinformatics. Casting the task into a classification problem makes it challenging, since the classes (functions) to be predicted are hierarchically related, and a protein can have more than one function. One approach is to produce a set of local classifiers; each is responsible for discriminating between a subset of the classes in a certain level of the hierarchy. In this paper we tackle the hierarchical classification problem in a local fashion, by learning an ensemble of Bayesian network classifiers for each class in the hierarchy and combining their outputs with four alternative methods: a) selecting the best classifier, b) majority voting, c) weighted voting, and d) constructing a meta-classifier. The ensemble is built using ABC-Miner, our recently introduced Ant-based Bayesian Classification algorithm. We use different types of protein representations to learn different classification models. We empirically evaluate our proposed methods on an ageing-related protein dataset created for this research.

Keywords: Ant Colony Optimization, Data Mining, Bayesian Networks, Hierarchical Classification, Protein Function Prediction.

1 Introduction

Data mining, a research focus in the fields of artificial intelligence, machine learning and statistics, is the process of discovering accurate, comprehensible and useful patterns in real-world datasets. Classification is one of the widely studied data mining tasks, in which the aim is to learn a model used to predict the class of unlabelled cases [22]. Many real-world classification problems have their classes organized into a hierarchy – typically a tree or a Directed Acyclic Graph (DAG), which makes the hierarchical classification problem a challenging research topic. A commonly used approach to tackle such a problem is the local approach, where the class hierarchy is processed in a top-down fashion, producing one or more local classifiers for each class level and combining their outputs. Each classifier is trained with a flat classification algorithm using a local data subset, and it discriminates among a subset of classes in the hierarchy.

L. Vanneschi, W.S. Bush, and M. Giacobini (Eds.): EvoBIO 2013, LNCS 7833, pp. 80–91, 2013.

Protein function prediction is an important application of hierarchical classification, and is considered as one of the major types of bioinformatics problems [7]. Determining protein functions is crucial for improving biological knowledge, diagnosis and treatment of diseases. In this work, we focus on human ageing-related proteins, predicting their biological processes according to the Gene Ontology. Research on ageing is important because ageing is the greatest risk factor for a number of diseases.

ABC-Miner [14], recently introduced by the authors, is a flat classification algorithm that learns the structure of a Bayesian Augmented Naïve-Bayes (BAN) network using Ant Colony Optimization (ACO)– a meta-heuristic global search for solving combinatorial optimization problems [6].

In this paper we approach the hierarchical protein function prediction in a local fashion, using the ABC-Miner algorithm. We build an ensemble of classifiers for each node in the class hierarchy using the same algorithm but with four different types of protein representations, which were the representation types used in [18]. Each representation consists of a predefined set of protein features of the same type. There are many types of protein representations, and the choice of the feature representation might be as important as the choice of the classification algorithm. The issue of using different representations is related to the well-known issue of feature selection in data mining, where a subset of the features is selected during the run of the algorithm to build a classifier. However, we favour the former, mainly because it significantly reduces computational time. This is important in our local hierarchical classification problem, where we have to build a classifier at each of the large number of class hierarchy nodes.

The combination of an ensemble's outputs at each class node is performed by four alternative methods: a) selecting the best classifier, b) majority voting, c) weighted voting, and d) constructing a meta-classifier to perform the final prediction. From one perspective, we compare the use of each protein representation individually to the use of the various proposed ensemble methods to combine representations. From another perspective, we compare the use of our ant-based algorithm – on each protein representation individually and with the various ensemble methods – to other well-known Bayesian classification algorithms, namely: Naïve-Bayes, TAN, and GBN. We evaluate the classification performance of our ensemble settings on a new dataset of ageing-related proteins, which was created for this research – due to the lack of ageing datasets for hierarchical classification in the literature.

The rest of the paper is organized as follows. Section 2 gives a brief overview on Bayesian network classifiers. Section 3 describes our Ant-based Bayesian Classification Algorithm. Section 4 discusses hierarchical classification approaches. Section 5 describes our proposed ensemble-of-classifier methods based on different protein representations for hierarchical classification. Section 6 describes the creation of our ageing-related dataset. Experimental results are shown in Section 7, followed by the conclusion and future research directions in Section 8.

2 Bayesian Network Classifiers

In the context of reasoning with uncertainty, Bayesian networks (BN) is one of the most powerful tools that model (in)dependence relationships between variables [5]. A directed acyclic graph (DAG) is used to represent the variables as nodes and statistical dependencies between the variables as edges between the nodes. In addition, a set of conditional probability tables (network parameters), one for each variable, is obtained by computing the probability distribution of the variable given its parents. Note that a Bayesian network should be able to answer probabilistic queries about any node(s) in the network.

Bayesian network classifiers are a special kind of probabilistic networks, where the concern is to answer queries about the probability of a specific node: the class attribute. Thus, the class node is treated as a special variable in the network; it is set as the parent of all other variables. The purpose is to compute the probability of each value c in the class variable C given a case $\mathbf{x}$ (an instance of the input attributes $\mathbf{X} = \{X_1, X_2, ..., X_n\}$), then label the case with the class having the highest probability, as in the following formulas:

$$C(\mathbf{x}) = \arg\max_{\forall c \in C} P(C = c | \mathbf{x} = x_1, x_2, ..., x_n), \tag{1}$$

letting $\mathbf{Pa}(X_i)$ be the set of parent predictor variables of X_i in the network, according to the Bayes' Theorem:

$$\overbrace{P(C = c | \mathbf{x} = x_1, x_2, ..., x_n)}^{\text{posterior probability}} \alpha \overbrace{P(C = c)}^{\text{prior probability}} \prod_{i=1}^{n} \overbrace{P(x_i | \mathbf{Pa}(X_i), C = c)}^{\text{likelihood}} \tag{2}$$

There are several types of Bayesian network classifiers studied in the literature: Naïve-Bayes, Tree Augmented Naïve-Bayes (TAN), Bayesian network Augmented Naïve-Bayes (BAN), and General Bayesian networks (GBN). Naïve-Bayes is the simplest Bayesian classifier in the literature, and it assumes the attributes are independent given the class label [3] – i.e., each feature node has just one parent node in the network: the class variable.

Since the independency assumption in general is not realistic, extended versions were developed to improve the performance of Naïve-Bayes. TAN allows a node in a BN to have more than one parent, besides the class variable. This produces a tree-like BN structure. In BAN classifiers, no restrictions (or at most k-dependencies) are enforced on the number of parents that a node in the network can depend on. Unlike the other BN classifier learners, a GBN learner treats the class variable node as an ordinary node during network construction. The idea is to build a general purpose BN, extract the *Markov blanket* of the class node and use the resulting network as a Bayesian classifier. For a review and comparison of various BN classifiers, see [3,9].

Learning a BN (classifier) from a dataset $\mathbf{D}$ is decomposed into two phases; learning the network structure and learning the network parameters. Parameter learning can be done in a relatively straightforward way by computating a

conditional probability table (CPT) for each variable with respect to its parent variables. The CPT of variable X_i encodes the likelihood of each value of this variable given each combination of values of $\mathbf{Pa}(X_i)$ in the network structure G, and the likelihood of the dataset $\mathbf{D}$ given a network G is denoted by $P(\mathbf{D}|G)$. Typically, the purpose is to find G that maximizes $P(\mathbf{D}|G)$ for a given $\mathbf{D}$, which is the role of BN structure learning. A common approach to this problem is to introduce a scoring function, f, that evaluates each G with respect to $\mathbf{D}$, searching for the best network structure according to this score [3].

Most algorithms used in the literature for building such BN classifiers are deterministic and greedy, and so are likely to get trapped into local optima in the search space. Since learning the optimal BN structure from a dataset is $\mathcal{NP}$-complete, stochastic meta-heuristic global search methods like ACO, which are less likely to get trapped into local optima, can be applied to build high-quality BN classifiers in an acceptable computational time.

3 The ABC-Miner Algorithm

Ant Colony Optimization (ACO) [6] is a meta-heuristic for solving combinatorial optimization problems, inspired by observations of the behavior of ant colonies in nature. The main idea is to utilize a swarm of simple individuals that use collective behaviour to achieve a certain goal. ACO algorithms have been successful in solving several combinatorial optimization problems, including classification rules discovery [12,11] and general purpose BN construction [2,13,23]. However, ABC-Miner [14], recently introduced by the authors, is the first ACO algorithm to learn BN classifiers.

In ABC-Miner the decision components in the construction graph (which define the search space that an ant uses to construct a candidate solution) are all the edges of the form $X \rightarrow Y$ where $X \neq Y$ and X, Y belong to the set of predictor attributes. These edges represent the attribute dependencies in a constructed BN classifier. Each ant in the colony creates a candidate solution (BN classifier). Then the quality of that solution is evaluated. The best solution produced in the colony at the current iteration is selected to undergo local search before the ant updates the pheromone trail on the construction graph according to the quality of its solution. The pheromone amounts deposited on the decision components guide the subsequent ants towards new better candidate solutions. After that, it compares the current iteration's best solution with the global best solution to keep track of the best solution found along the entire search so far. This set of steps is repeated until the algorithm converges on a solution or the maximum number of iterations is reached.

In order to build the structure of a BN classifier, the maximum number of parents for a node is typically specified by the user. However, the selection of the optimal number of dependencies that a variable in the network can have is automatically carried out in ABC-Miner [14]. To create a candidate solution, an ant starts with the network structure of Naïve-Bayes, i.e. a BN in which all the variables have only the class variable as the parent. Then it expands that

structure into a Bayesian Augmented Naïve-Bayes (BAN) structure by adding edges to the network. The selection of the edges is performed according to a probabilistic state transition formula that involves pheromone amount and the heuristic information – using conditional mutual information [14] associated with the edges. An edge is valid to be added to the BN classifier being constructed if its inclusion does not create a directed cycle and does not exceed the limit of k parents (chosen by the current ant). After the ant adds a valid edge, all the invalid edges are eliminated from the construction graph. The ant keeps adding edges to the current solution until no valid edges are available. When the structure is finished, the CPT is computed for each variable, producing a complete BN classifier. Then the quality of the solution is evaluated and all the edges become available for constructing further candidate solutions.

The ABC-Miner algorithm evaluates the quality of the BN classifier using *accuracy* [14], a conventional measure of predictive performance, since the goal is to build a BN only for predicting the value of a specific class attribute, unlike conventional BN learning algorithms whose scoring function do not distinguish between the predictor and the class attributes. Experimental evaluations showed the predictive effectiveness of ABC-Miner in flat classification comparing to other Bayesian classification algorithms, namely: Naïve-Bayes, TAN and GBN [14]. Thus, we carry on using it for hierarchical classification.

4 Hierarchical Classification

Hierarchical classification refers to the task of predicting the class value(s) of a given case in a domain where the class values are arranged into a hierarchy. Many real-world classification problems have hierarchical classes, were a case belongs to a series of classes related in a general-to-specific structure. Such class structure is found, e.g., in document topics, music genres and protein functions, which makes the classification task more complex and challenging [19].

Figure 1 shows examples of hierarchical classification problems. A hierarchical classification problem can be characterized by three properties:

Graph Structure - This specifies whether the type of graph representing the hierarchical classes is tree or DAG. A node in the tree structure has only one parent, while it can have multiple parents in the DAG.
Labelling Type - This indicates whether a case is allowed to have class labels associated with a single or multiple paths in the class hierarchy. The hierarchy can be a tree, yet a case can be labelled with classes in different paths.
Labelling Depth - In full depth labelling, every case is labelled with classes at all levels, from the root to the leaf level. Partial depth labelling indicates that for some cases the value of the class label at the leaf level is not specified.

There are three different broad approaches to tackle hierarchical classification problems [19]. The first (and the simplest) approach is to completely ignore the class hierarchy and convert the problem into flat classification by predicting only classes at the leaf nodes. The second approach is to produce one or

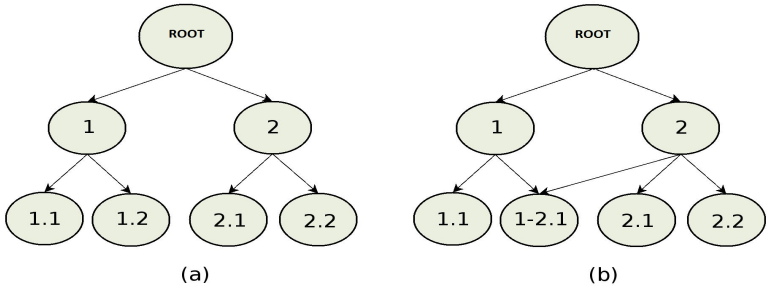


Fig. 1. Examples of hierarchies of classes: (a) Tree and (b) DAG. If each case in (a) is labelled with exactly one of the classes: 1.1, 1.2, 2.1, 2.2, then (a) is a single-path and full-depth classification problem. If a case in (b) is labelled with only class 2, and another case is labelled with both 1.1 and 1-2.1, then (b) is a multiple-path and partial-depth problem.

more local classifiers for each class level and combine their outputs. Each classifier is trained with a flat classification algorithm using a local data subset, discriminating among a subset of classes in the hierarchy. The third (and most complex) approach is to design a specific classification algorithm that can process the whole hierarchy in a global way. The local approach is widely used since it utilizes existing flat classification algorithms and the divide-and-conquer principle (see below) to solve more easily manageable classification problems, by comparison with the first and third approaches where a single classifier has to discriminate among a very large set of all leaf classes or all classes in the hierarchy, respectively.

The local approach for hierarchical classification uses the principle of "divide and conquer" to process the class hierarchy in a top-down fashion. More precisely, we use the "one-classifier-per-node" version of the local approach, which is suitable for our dataset, where the graph type is DAG, labelling type is multiple paths, and labelling depth is partial depth (see Section 6). In this local approach version, a binary flat classifier is built for each class in the hierarchy (except the root node). The flat classification algorithm at a given node uses the cases belonging to this class node as positive examples, and the cases belonging to the siblings of this class node (but not to the current class node) as negative examples in the training phase. To illustrate how to classify a case in the testing phase, we use example (b) in Figure 1. The local classifier at node 1 decides whether a given case belongs to this class or not, the same for node 2. If the case is classified as belonging to class 2, it is passed to the classifiers of the class nodes 2.1 and 2.2 (the case is also passed to the classifier at node 1-2.1 if it is classified as belonging to class 1 and 2) at the next level. If the case does not belong to class 2, no further classifications are performed from node 2. In this example, a case can be labelled with, say, classes 1, 2, and 2.2, which is appropriate for the aforementioned characteristics of our dataset.

5 Proposed Methods for an Ensemble of Classifiers

An ensemble of classifiers is often used to combine the predictions from separate classifiers in order to increase predictive accuracy. The idea is to construct an ensemble of classifiers having different inductive biases, so they make different errors. Hence, combining their classification outputs will make the overall prediction of the ensemble more accurate [22].

There are two main issues in ensembles of classifiers. First, the type of diversity in the classifiers: different algorithms with the same dataset, different attributes (features) from the same dataset, different data representations with the same algorithm, and different training case subsets with the same algorithm (e.g., bagging). The second issue is how the classes predicted by the classifiers are combined. Table 1 shows how our work fits into the context of related work on hierarchical classification according to these two issues.

Table 1. Related Work on Ensembles in Hierarchical Protein Function Classification

		Approaches to Combine Classifier Ensemble' Predictions			
		Select the best	Majority Vote	Weighted Vote	Stacking
Diversity in the Ensemble	Different Algorithms	Secker et al. [17], Silla et al. [18]			Costa et al. [4]
	Different Data Representations	**This Work**, Silla et al. [18]	**This Work**	**This Work**	**This Work**
	Different Feature Subsets	Secker et al. [16]			
	Different Data subsets			Schietgat et al. [15]	

In addition to the use of our ABC-Miner for hierarchical protein function prediction in a new ageing-related dataset, what is novel in this work are the various methods we employ to combine the outputs of an ensemble's classifiers built with different protein representations in hierarchical classification. As shown in Table 1, three of the four cells marked with the keyword "this work" involve a new combination of the technique of building classifiers with different protein representations (proposed in [18]) with a technique for combining the classifiers' predictions.

In essence, ABC-Miner builds an ensemble of four binary classifiers (one for each protein representation, i.e. a pre-defined feature set) for each class node in the hierarchy. The selective method chooses the best classifier – with the highest accuracy in a validation set (a subset of the training set) [17] – in the ensemble to predict whether or not a given case belongs to the current local class node. The majority voting method uses the majority decision of the ensemble to make a local class prediction. For example, if three out of four classifiers predicted

that a given case does not belong to the current class, this case is not labelled by this class. The weighted voting method weights the vote of each classifier by its accuracy on a validation set (again, a subset of the training set), and predicts the class with the the largest sum of weighted votes.

The stacking method is a meta-classifier built upon the predicted classes of the ensemble's classifiers to perform the final classification of a case. For a given training case, a meta-case is constructed with four attributes: each represents the class predicted by one of the ensemble's four classifiers. The class of the meta-case is the same as the one in the training case. At each node in the hierarchy the algorithm uses an ensemble's classifiers to classify the n cases of its local training subset and then n meta-cases are generated upon the output of the classifiers. A meta-classifier is built to learn the relationships between the predictions of the various classifiers and the actual classes. In the testing phase, the meta-classifier outputs the final classification for a test case upon the predictions of the ensemble's classifiers.

The accuracy of a classifier in an ensemble is measured by the hierarchical F-measure, which combines hierarchical precision (hP) and hierarchical recall (hR), as shown in the following formula, where A_i is the set of actual (true) class labels and P_i is the set of predicted class labels, respectively, for the ith case, and n is the total number of validation cases.

$$hP = \frac{\sum_{i=1}^{n} |A_i \cup P_i|}{\sum_{i=1}^{n} |P_i|} \quad hR = \frac{\sum_{i=1}^{n} |A_i \cup P_i|}{\sum_{i=1}^{n} |A_i|} \quad hF = \frac{2.hP.hR}{hP + hR}\,, \tag{3}$$

6 The Ageing-Related Protein Dataset

A new dataset of human ageing-related protein was created specifically to be used in our experiments. A case (protein) has four different sets of predictor attributes (four protein representations), extracted from its amino acid sequence as described later, and a set of class labels (functions) to be predicted. These functions are hierarchically-related biological processes in the Gene Ontology (GO) [20]. To create the dataset, first a set of ageing genes were obtained from the GenAge database [10], which contained 260 human ageing-related genes (species "Homo sapiens"). Note that we obtained only the gene names from the GenAge dataset, the proteins target classes and features were obtained as follows.

Next, we used the Swiss-Prot database [21] from the UniProt knowledge base to obtain the sequences of the proteins corresponding to those genes (from which the predictor attributes are extracted) and the biological process GO terms (to be predicted) for each protein. The total number of GO terms included in that set of proteins was 2889. Considering each of those terms as a class to be predicted is not desirable, for two reasons. First, the number of classes in the hierarchy would be very large, with many classes having very few cases (making their prediction unreliable). Second, a lot of those GO terms are irrelevant to the process of ageing, since an ageing-related protein often performs other biological processes unrelated to ageing. Therefore, it is desirable to have a subset of these terms containing only GO terms related to the ageing process.

In order to mitigate these problems, we used the DAVID [8] bioinformatics tool to identify *enriched* GO terms in our set of ageing-related proteins. DAVID performs statistical tests to identify the GO terms whose annotations are most correlated with a specific set of proteins. We set the minimum number of proteins per GO term to 15, and set the EASE parameter that represents the maximum p-value of the enrichment significance to 1E-10. We obtained 102 biological process GO terms, which were used as the class hierarchy in our dataset. The dataset was reduced to 247 proteins, as 16 were lost due to the GO terms reduction. Table 2 shows the top 100 GO terms (GO:Term Proteins_Count) in our dataset sorted by the protein count. The top left term is the root node that has the 247 proteins. To see the names and the hierarchical organization of the terms, please refer to QuickGO [1], where only the "IS-A" relationship is considered in our construction of the class hierarchy.

Table 2. Top 100 GO Terms in the Dataset and Their Protein Counts

GO:0008150 247	GO:0034641 119	GO:0031325 91	GO:0043085 62	GO:0048856 42
GO:0009987 241	GO:0007165 119	GO:0010604 89	GO:0050793 61	GO:0009725 40
GO:0065007 207	GO:0009889 119	GO:0032502 85	GO:0009892 60	GO:0010942 40
GO:0050789 204	GO:0051171 118	GO:0007166 85	GO:0010557 60	GO:0043068 40
GO:0050794 201	GO:0031326 116	GO:0065008 84	GO:0006259 59	GO:0043065 40
GO:0008152 196	GO:0019219 116	GO:0033554 78	GO:0031399 59	GO:0045944 40
GO:0044237 188	GO:0048519 115	GO:0050790 78	GO:0051726 56	GO:0051094 38
GO:0050896 184	GO:0006950 111	GO:0010033 76	GO:0006974 55	GO:0009628 37
GO:0044238 181	GO:0042221 111	GO:0010941 76	GO:0010605 55	GO:0006281 37
GO:0051716 163	GO:0006139 110	GO:0043067 76	GO:0006357 54	GO:0008284 37
GO:0019222 158	GO:0048523 110	GO:0042981 76	GO:0007167 53	GO:0012502 31
GO:0043170 156	GO:0010556 107	GO:0035556 75	GO:0051254 53	GO:0006917 31
GO:0044260 150	GO:0090304 104	GO:0044093 74	GO:0010628 52	GO:0009314 29
GO:0031323 148	GO:2000112 104	GO:0051246 72	GO:0007169 51	GO:0040008 28
GO:0080090 145	GO:0010468 103	GO:0009891 69	GO:0060548 49	GO:0043434 27
GO:0048518 144	GO:0065009 102	GO:0032268 69	GO:0043069 49	GO:0051052 27
GO:0060255 143	GO:0009893 97	GO:0051173 67	GO:0045893 49	GO:0006979 26
GO:0048522 135	GO:0051252 96	GO:0031328 67	GO:0043066 49	GO:0006916 26
GO:0006807 120	GO:2001141 96	GO:0045935 64	GO:0009719 47	GO:0048513 19
GO:0034641 119	GO:0006355 94	GO:0042127 62	GO:0042592 44	GO:0009416 19

We extracted 4 sets of protein features (used in [18]), each set constituting a different protein representation, from the proteins' amino acid sequences. In essence, the first feature set contains 5 z-values, representing hydrophobicity/hydrophilicity (z1), steric/bulk properties and polarizability (z2), polarity (z3), and electronic effects (z4 and z5) of the amino acids in the whole sequence. The second set contains 15 z-values; 5-values are averaged over the whole sequence, other 5 z-values are averaged over the N-terminus (the first 150 amino acids) of the protein, and further 5 z-values are computed from the C-terminus (the last 150 amino acids) of the sequence.

The third feature set contains amino acid compositions, which are the percentages of occurrence of each amino acid within a protein sequence. This produces a set of 20 features, each of them with the percentage of how many times a given amino acid occurs within the protein sequence. The fourth set contains local descriptors, consisting of 21 features (3 Composition, 3 Transition and 15 Distribution features), which are computed based on the variation of occurrence of functional groups of amino acids within the protein sequence. The functional groups used were: hydrophobic (amino acids CVLIMFW), neutral (amino acids GASTPHY), and polar (amino acids RKEDQN). We also added the sequence length and molecular weight of the protein as two additional features to each of the feature sets, since both are easy to obtain and may be somewhat relevant to protein functional prediction.

7 Experimental Results

The experiments were carried out using a well-known 5-fold cross validation procedure [22]. The results (average F-measure on the test set) are reported in Table 3. From one perspective, we compare the performance of the various ensemble-of-classifiers methods to the use of each protein representation separately. From another perspective, we compare the use of ABC-Miner as the base classifier in an ensemble to the use of other BN classification algorithms: Naïve-Bayes, GBN and TAN. Parameter settings are the same as in [14].

In Table 3, the top four rows refer to the use of each protein representation separately, so that no ensemble of classifiers is built. The bottom four rows refer to experiments with ensembles, using the various methods of prediction combination (discussed in Section 5) of the ensemble of classifiers built with different representations. An entry in bold face indicates that it is the highest F-measure value obtained across the four algorithms for the same experiment variation. An underlined entry indicates that it is the highest F-measure value obtained in all of the experiment variations for the same algorithm.

Table 3. Predictive Performance (*mean* ± *standard error*) Results - F-measure

Experiment Variation	Naïve-Bayes	TAN	GBN	ABC-Miner
5 Z-values	0.196 ± 0.02	0.285 ± 0.01	0.218 ± 0.02	**0.342 ± 0.02**
15 Z-values	0.307 ± 0.01	**0.321 ± 0.01**	0.242 ± 0.01	0.298 ± 0.01
AA Composition	0.279 ± 0.01	0.283 ± 0.02	0.271 ± 0.01	**0.363 ± 0.02**
Local Descriptors	0.345 ± 0.00	0.340 ± 0.00	0.342 ± 0.00	**0.347 ± 0.00**
Selective Classifier	0.332 ± 0.01	0.332 ± 0.01	0.287 ± 0.01	**0.351 ± 0.02**
Majority Voting	0.269 ± 0.01	**0.372 ± 0.02**	0.251 ± 0.02	0.366 ± 0.01
Weighted Voting	0.376 ± 0.01	0.378 ± 0.02	0.376 ± 0.02	**0.481 ± 0.01**
Meta-Classifier	0.308 ± 0.02	0.381 ± 0.01	0.328 ± 0.02	**0.397 ± 0.02**

As shown in Table 3, ABC-Miner obtained the highest performance in 6 out of 8 experiments variations, where in 5 variations it was significantly better than the second best algorithm according to a two-tailed Student's t-test with significance level of 5%. TAN performed the second best, since it came in the first place in 2 variations and in the second place in 5 variations. In addition, the different approaches for combining an ensemble's classifiers' predictions have generally performed better than the use of each protein representation separately. Specifically, the weighted voting approach obtained the best results with 3 out of 4 algorithms. The meta-classifier came in the first place for 1 algorithm and in the second place for 2 algorithms. Unexpectedly, the selective approach has outperformed the majority vote for 2 algorithms. For each of the four used algorithms, the best ensemble method outperformed the best single protein representation at the statistical significance level of 5%, showing the effectiveness of the proposed ensemble methods over other approaches.

8 Concluding Remarks

In this paper we have used our recently introduced ant-based Bayesian classification algorithm in several ensembles of classifiers to tackle the hierarchical protein function prediction problem in a local approach. We created a new ageing-related protein dataset to carry out our experiments. Results have showed the effectiveness of the proposed ensemble methods in hierarchical classification, especially when ABC-Miner is used. However, lack of prediction model comprehensibility is an issue; a large number of classifiers (four for each class in the hierarchy) are built, reducing the interpretability of the models by users. This is an inherited drawback of the local approach. Therefore, an important research direction is to extend ABC-Miner to tackle hierarchical classification in a global fashion, where only one classification model is produced for the entire class hierarchy.

Acknowledgements. The authors thank Dr. Carlos Silla for extracting the feature sets used in our experiments and Dr. Joao Pedro de Magalhaes for his valuable advice about the creation of the ageing-related protein's dataset.

References

1. Binns, D., Dimmer, E., Huntley, R., Barrell, D., O'Donovan, C., Apweiler, R.: QuickGO: a web-based tool for Gene Ontology searching. Bioinformatics 25, 3045–3046 (2009)
2. de Campos, L.M., Fernandez-Luna, J.M., Gamez, J.A., Puerta, J.M.: Ant colony optimization for learning Bayesian networks. International Journal of Approximate Reasoning 31(3), 291–311 (2002)
3. Cheng, J., Greiner, R.: Learning Bayesian Belief Network Classifiers: Algorithms and System. In: Stroulia, E., Matwin, S. (eds.) AI 2001. LNCS (LNAI), vol. 2056, pp. 141–151. Springer, Heidelberg (2001)

4. Costa, E.P., Lorena, A.C., Carvalho, A.C.P.L.F., Freitas, A.A.: Top-Down Hierarchical Ensembles of Classifiers for Predicting G-Protein-Coupled-Receptor Functions. In: Bazzan, A.L.C., Craven, M., Martins, N.F. (eds.) BSB 2008. LNCS (LNBI), vol. 5167, pp. 35–46. Springer, Heidelberg (2008)
5. Daly, R., Shen, Q., Aitken, S.: Learning bayesian networks: Approaches and issues. Knowledge Engineering Reviews 26(2), 99–157 (2011)
6. Dorigo, M., Stützle, T.: Ant Colony Optimization. The MIT Press (2004)
7. Freitas, A.A., de Carvalho, A.C.P.F.L.: A tutorial on hierarchical classification with applications in bioinformatics. In: Research and Trends in Data Mining Technologies and Applications, pp. 175–208 (2007)
8. Huang, D., Sherman, B., Lempicki, R.: Systematic and integrative analysis of large gene lists using DAVID Bioinformatics Resources. Nature Protocol 4, 44–57 (2009)
9. Jiang, L., Wang, D., Cai, Z., Yan, X.: Survey of Improving Naive Bayes for Classification. In: Alhajj, R., Gao, H., Li, X., Li, J., Zaïane, O.R. (eds.) ADMA 2007. LNCS (LNAI), vol. 4632, pp. 134–145. Springer, Heidelberg (2007)
10. de Magalhaes, J., Budovsky, A., Lehmann, G., Costa, J., Li, Y., Church, V.F.G.: The Human Ageing Genomic Resources: online databases and tools for biogerontologists. Aging Cell, 65–72 (2009)
11. Martens, D., Baesens, B., Fawcett, T.: Editorial survey: swarm intelligence for data mining. Machine Learning 82(1), 1–42 (2011)
12. Parpinelli, R.S., Lopes, H.S., Freitas, A.A.: Data mining with an ant colony optimization algorithm. IEEE TEC 6, 321–332 (2002)
13. Pinto, P.C., Nägele, A., Dejori, M., Runkler, T.A., Sousa, J.M.C.: Using a local discovery ant algorithm for Bayesian network structure learning. IEEE Transactions on Evolutionary Computation 13(4), 767–779 (2009)
14. Salama, K.M., Freitas, A.A.: ABC-Miner: An Ant-Based Bayesian Classification Algorithm. In: Dorigo, M., Birattari, M., Blum, C., Christensen, A.L., Engelbrecht, A.P., Groß, R., Stützle, T. (eds.) ANTS 2012. LNCS, vol. 7461, pp. 13–24. Springer, Heidelberg (2012)
15. Schietgat, L., Vens, C., Struyf, J., Blockeel, H., Kocev, D., Dzeroski, S.: Predicting gene function using hierarchical multi-label decision tree ensembles. BMC Bioinformatics 11(1) (2010)
16. Secker, A., Davies, M.N., Freitas, A.A., Clark, E., Timmis, J., Flower, D.R.: Hierarchical classification of GPCR with data-driven selection of attributes and classifiers. International Journal of Data Mining and Bioinformatics 4(2), 191–210 (2010)
17. Secker, A., Davies, M.N., Freitas, A.A., Timmis, J., Mendao, M., Flower, D.R.: An experimental comparison of classification algorithms for the hierarchical prediction of protein function. Expert Update (BCS-SGAI Magazine) 9, 17–22 (2007)
18. Silla, C.N., Freitas, A.A.: Selecting different protein representations and classification algorithms in hierarchical protein function prediction. Intelligent Data Analysis 15(6), 979–999 (2011)
19. Silla, C.N., Freitas, A.A.: A survey of hierarchical classification across different application domains. Data Mining and Knowledge Discovery 22(1-2), 31–72 (2011)
20. The Gene Ontology Consortium: Gene ontology: tool for the unification of biology. Nature Genetics 25(1), 25–29 (2000)
21. The UniProt Consortium: The Universal Protein Resource (Uniprot). Nucleic Acids Research 38, D142–D148 (2010)
22. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 3rd edn. Morgan Kaufmann, San Francisco (2010)
23. Wu, Y., McCall, J., Corne, D.: Two novel Ant Colony Optimization approaches for Bayesian network structure learning. In: International Conference on Evolutionary Computation (CEC), pp. 1–7 (2010)

Dimensionality Reduction via Isomap with Lock-Step and Elastic Measures for Time Series Gene Expression Classification

Carlotta Orsenigo and Carlo Vercellis

Dept. of Management, Economics and Industrial Engineering, Politecnico di Milano, Via Lambruschini 4b, 20156 Milano, Italy

Abstract. Isometric feature mapping (Isomap) has proven high potential for nonlinear dimensionality reduction in a wide range of application domains. Isomap finds low-dimensional data projections by preserving global geometrical properties, which are expressed in terms of the Euclidean distances among points. In this paper we investigate the use of a recent variant of Isomap, called double-bounded tree-connected Isomap (dbt-Isomap), for dimensionality reduction in the context of time series gene expression classification. In order to deal with the projection of temporal sequences dbt-Isomap is combined with different lock-step and elastic measures which have been extensively proposed to evaluate time series similarity. These are represented by three $\mathcal{L}_p$-norms, dynamic time warping and the distance based on the longest common subsequence model. Computational experiments concerning the classification of two time series gene expression data sets showed the usefulness of dbt-Isomap for dimensionality reduction. Moreover, they highlighted the effectiveness of $\mathcal{L}_1$-norm which appeared as the best alternative to the Euclidean metric for time series gene expression embedding.

Keywords: Dimensionality reduction, isometric feature mapping, time series similarity measures, time series gene expression classification.

1 Introduction

In the last decade several efforts have been devoted to the development of classification methods for high-dimensional gene expression data originated by microarray experiments. Most of these studies rely on static data sets, which collect the expression levels of thousands of genes observed simultaneously under specific experimental conditions.

In order to model the complex interactions of biological processes, however, it is often required to analyze the evolution of gene expression profiles over time. In functional genomics, for example, the classification of gene expression patterns recorded across a series of time points plays a prominent role since genes with similar profiles are supposed to be functionally related or co-regulated [1]. In toxicogenomics the analysis of gene expression changes with signal pathway activation may provide insight into the mechanism of action of a chemical, and

L. Vanneschi, W.S. Bush, and M. Giacobini (Eds.): EvoBIO 2013, LNCS 7833, pp. 92–103, 2013.

deliver specific information about its biological effects [2]. As a third example, consider the prediction of the clinical response to a drug [3], where different rates of disease development or treatment response may be observed among patients. In this case, gene expression profiles may be similar but misaligned since the disease may progress at different speed for different individuals. All these applications call for the development of effective methods for analyzing time series microarray measurements.

In the context of dimensionality reduction several linear and nonlinear approaches have been proposed. Classical methods addressing linear data projections include principal component analysis [4] and metric multidimensional scaling (MDS) [5]. Among nonlinear techniques, which are capable of handling data with complex nonlinear structures, manifold learning methods have attracted great attention. They include isometric feature mapping (Isomap) [6], locally linear embedding [7] and Laplacian eigenmaps [8], among others. Manifold learning methods attempt to recover the low-dimensional manifold along which data are supposed to lie. Given a set of points $\mathcal{S}_m = \{\mathbf{x}_i,\ i \in \mathcal{M} = \{1, 2, \ldots, m\}\} \subset \Re^n$ arranged along a nonlinear manifold M of unknown dimension d, with $d \ll n$, they aim at finding a function $f : M \to \Re^d$ mapping $\mathcal{S}_m$ into $\mathcal{D}_m = \{\mathbf{z}_i,\ i \in \mathcal{M} = \{1, 2, \ldots, m\}\} \subset \Re^d$ such that some geometrical properties of the data in the input space are preserved in the projection space. In the last decade manifold learning techniques have been successfully applied in a wide range of application domains for both unsupervised tasks and supervised learning. An empirical comparison of these methods for cancer microarray data classification is presented in [9].

In this paper we investigate the use of a recent variant of isometric feature mapping, called double-bounded tree-connected Isomap (dbt-Isomap), for nonlinear dimensionality reduction in the context of time series gene expression classification. In order to deal with the embedding of temporal sequences, dbt-Isomap is alternatively combined with lock-step and elastic distance measures, represented by three $\mathcal{L}_p$-norms, dynamic time warping and the distance based on the longest common subsequence model. Our aim was to determine whether low-dimensional time series projection may benefit from the use of distance functions which have been extensively proposed to properly grasp the similarity among sequences, in the prospect of inducing better performance in the subsequent classification task. Time series embedding approaches combining MDS and Laplacian eigenmaps with dynamic time warping were proposed in [10]. We are not aware of any previous studies on dimensionality reduction resorting to Isomap with alternative distance metrics for time series gene expression classification.

The remainder of the paper is organized as follows. Section 2 offers an overview of Isomap and of its double-bounded tree-connected variant. Section 3 briefly describes the time series similarity measures considered in this study. Section 4 illustrates the experimental settings and the comparative results concerning the classification of two gene expression data sets. Conclusions and future extensions are discussed in section 5.

2 Double-Bounded Tree-Connected Isomap

Isometric feature mapping is a generalization of metric multidimensional scaling to nonlinear manifolds. Classical MDS attempts to preserve the Euclidean distance between data points in the input and in the low-dimensional space. On the other hand, Isomap finds an embedding in which the geodesic distance between two points in the original space is as close as possible to the Euclidean distance between their projections in the target space. The geodesic distance is defined by the length of the shortest curve connecting two points on the manifold. This is estimated by the shortest path between the corresponding vertices in a weighted neighborhood graph, in which every point is connected to its k nearest neighbors and the weight of an edge is given by the Euclidean distance between its endpoints. Notice that, the shortest path between any pair of vertices can be computed only when the neighborhood graph is connected. The embedding in the low-dimensional space is therefore obtained by the singular value decomposition of the squared geodesic distances matrix.

The original Isomap algorithm relies on two alternative criteria for building the neighborhood of each point: searching for the k nearest neighbors or selecting all points contained in an ε-radius hypersphere centered on the point. In both cases, the value of k or ε should be large enough in order to guarantee the connection of the neighborhood graph.

As observed in [11], the k nearest neighbors or the ε-radius rule may generate short-circuits within the manifold that cause topological instability, especially when the data set is affected by noise, outliers or scarcity of examples. To overcome this drawback a variant of Isomap, called double-bounded tree-connected Isomap (dbt-Isomap), was recently proposed in [12]. dbt-Isomap is capable of preventing short-circuits by combining the aforementioned criteria. Precisely, for building the neighborhood graph it searches for at most k nearest neighbors within an ε-radius hypersphere around each point. If the graph turns out to be disconnected the separate subgraphs are then joined by generating a minimum spanning tree among the points that best represent the centroids of the single components. This always results in a connected graph without unnaturally increasing the values of k and ε.

Isomap has been successfully applied to the analysis of high-dimensional biomedical data [13, 14]. In the context of classification its double-bounded variant has proven to be more accurate and robust with respect to the original algorithm [12]. For this reason, in the present study we confined the attention to dbt-Isomap for nonlinear dimensionality reduction. In particular, in light of the nature of the classification task which involves temporal sequences, we modified dbt-Isomap by combining it with alternative distance measures for building the neighborhood graph. The distance between two points is, in this case, intimately related to the similarity of the corresponding time series. Therefore, the creation of the neighborhood graph may benefit from the use of distance metrics which have been proposed as effective time series similarity measures. In its turn, this may positively affect the quality of the data embedding and enhance the performance in the subsequent classification stage.

The dbt-Isomap algorithm applied in this study can be summarized as follows, where *Dist* indicates the specific metric used for computing the distance between any pair of points.

Procedure dbt-Isomap($\mathcal{S}_m, d, \varepsilon, k, Dist$)

1. Build the neighborhood graph by connecting each point of $\mathcal{S}_m$ to at most k nearest neighbors within an ε-radius hypersphere. The distance between two points is computed by means of the *Dist* function.
2. Find the T connected components of the neighborhood graph. If $T > 1$:
   (a) for each component compute the centroid defined as the average of all the points in the component;
   (b) for each component choose the point that is closest to the corresponding centroid; these points are called *approximate centroids* of the components.
   (c) find a minimum spanning tree in the complete graph composed by the approximate centroids.
3. Estimate the geodesic distances as the length the shortest paths computed between every pair of vertices in the neighborhood graph.
4. Find the embedding in the low d-dimensional space by applying classical MDS algorithm. To this aim, first compute the square m-dimensional matrix $\mathbf{K} = -\mathbf{HSH}/2$, where $\mathbf{S}$ is the matrix of squared geodesic distances and $\mathbf{H}$ is the centering matrix of size m. Then, consider the square d-dimensional diagonal matrix $\mathbf{\Lambda}$ composed by the first d largest eigenvalues of $\mathbf{K}$ and the $m \times d$ matrix $\mathbf{V}$ of associated eigenvectors. Finally, find the embedding of the points as $\mathbf{Z} = \mathbf{V}\mathbf{\Lambda}^{1/2}$, where $\mathbf{Z}$ is a $m \times d$ matrix whose rows are the projections $\mathbf{z}_i, i \in \mathcal{M}$, of the points in the low d-dimensional space.

Isomap algorithms require an embedding procedure for mapping new data points in the low-dimensional space. To this aim, one may effectively resort to general regression neural networks [15] or to multi-output kernel ridge regression [16]. Furthermore, the intrinsic dimensionality d of the embedding space can be identified by retaining the dimension at which the curve of residual variance flattens. The residual variance is defined by $1 - R^2(\hat{\mathbf{D}}_M, \mathbf{D}_Z)$, where $\hat{\mathbf{D}}_M$ is the geodesic distance matrix, $\mathbf{D}_Z$ is the matrix of Euclidean distances in the low-dimensional space and R denotes the linear correlation coefficient over all entries of $\hat{\mathbf{D}}_M$ and $\mathbf{D}_Z$. Notice that, provided with enough data, isometric feature mapping is guaranteed to recover the true dimensionality of a large class of nonlinear manifolds [6].

3 Time Series Similarity Measures

This section provides a brief description of the time series similarity measures considered in this study. Our attention was confined to two families of distance functions which can be referred to as *lock-step* and *elastic* measures, respectively [17]. Lock-step measures, such as the Euclidean distance and other $\mathcal{L}_p$-norms, compute the similarity between two time series by matching each pair

of corresponding data points along the time axes. Elastic measures, which comprise dynamic time warping and longest common subsequence distance, allow to compare one data point of the first times series to multiple data points of the second sequence, in order to determine a coherent alignment and identify similar profiles with different phases.

It has been observed that for time series classification the Euclidean distance is competitive with the elastic measures when the size of the data set is large. For small data sets, elastic measures usually provide better performance with respect to the lock-step counterparts [17].

3.1 $\mathcal{L}_p$-Norm Distances

The most common and intuitive similarity measures for temporal sequences are derived from the $\mathcal{L}_p$-norm distance [18], which is also known as Minkowski distance of order p.

Given a pair of time series $\mathbf{x}_i, \mathbf{x}_k \in \Re^n$, the $\mathcal{L}_p$-norm distance is defined as

$$\mathcal{L}_p\left(\mathbf{x}_i, \mathbf{x}_k\right) = \left\|\mathbf{x}_i - \mathbf{x}_k\right\|_p = \left(\sum_{t=1}^{n} \left|\mathbf{x}_i\left(t\right) - \mathbf{x}_k\left(t\right)\right|^p\right)^{1/p}, \tag{1}$$

where $p \in [1, \infty]$ and $\mathbf{x}_i\left(t\right)$ is the sequence value at time t. When $p = 1$ the Manhattan distance is obtained. For $p = 2$, equation (1) defines the Euclidean distance. In the limiting case of $p = \infty$, the $\mathcal{L}_\infty$-norm, which is also called Maximum norm, is given by

$$\mathcal{L}_\infty\left(\mathbf{x}_i, \mathbf{x}_k\right) = \max_{t=1}^{n} \left|\mathbf{x}_i\left(t\right) - \mathbf{x}_k\left(t\right)\right|. \tag{2}$$

Distance based on $\mathcal{L}_p$-norms are parameter-free, easy to implement and interpret. By comparing the corresponding values for a given time point, however, they are sensitive to noise and may fail to properly catch sequential patterns similarity when these exhibit misalignments in time [17].

3.2 Dynamic Time Warping

Dynamic time warping (DTW) was originally proposed in the context of speech recognition and signal processing [19] and has been successfully applied for time series clustering, labeling and classification [20–22]. Given a pair of time series $\mathbf{x}_i \in \Re^n$ and $\mathbf{x}_k \in \Re^s$, dynamic time warping is defined by

$$\begin{aligned} DTW\left(\mathbf{x}_i, \mathbf{x}_k\right) = {} & \mathcal{L}_p\left(\mathbf{x}_i\left(n\right), \mathbf{x}_k\left(s\right)\right) + \\ & \min\left\{DTW\left(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_k\right), DTW\left(\hat{\mathbf{x}}_i, \mathbf{x}_k\right), DTW\left(\mathbf{x}_i, \hat{\mathbf{x}}_k\right)\right\}, \end{aligned} \tag{3}$$

where $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{x}}_k$ denote the sequences $\hat{\mathbf{x}}_i = \left(\mathbf{x}_i\left(1\right), \ldots, \mathbf{x}_i\left(n-1\right)\right)$ and $\hat{\mathbf{x}}_k = \left(\mathbf{x}_k\left(1\right), \ldots, \mathbf{x}_k\left(s-1\right)\right)$, respectively, and $\mathcal{L}_p$ is any p-norm. In practice, $\mathcal{L}_p$ is usually set to the squared Euclidean distance.

To speed up DTW evaluation it is customary to enforce some form of local or global constraints which are generally expressed in terms of a threshold δ on the warping window size, in order to prevent the alignment of points which are too far from each other along the time axes. It has been noticed [23] that these constraints often lead to more accurate similarity measures, beyond an improvement in the computational efficiency. Using dynamic programming and constraining the window size within δ, the time required to compute dynamic time warping is $O(\delta(n+s))$ [24].

Unlike $\mathcal{L}_p$-norm distances, DTW is able to deal with time series of variable length. Furthermore, as a similarity measure it has proven to be more robust and flexible. Indeed, it allows to match a point of one time series to one or many points of the other in order to perform shifts in the sequences and identify similar profiles in the presence of nonlinear misalignments.

3.3 Longest Common Subsequence Distance

An alternative approach is based on the use of the longest common subsequence (LCS) model, for which the similarity of two time series is related to the length of their longest common subsequence [24].

Given a pair of time series $\mathbf{x}_i \in \Re^n$ and $\mathbf{x}_k \in \Re^s$, the longest common subsequence is given by

$$LCS_{\delta,\gamma}(\mathbf{x}_i, \mathbf{x}_k) = \begin{cases} 0 \;\; if\ \mathbf{x}_i = \mathbf{0}\ or\ \mathbf{x}_k = \mathbf{0} \\ 1 + LCS_{\delta,\gamma}(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_k) \;\; if\ |\mathbf{x}_i(n) - \mathbf{x}_k(s)| < \gamma\ and\ |n-s| \leq \delta \\ \max\{LCS_{\delta,\gamma}(\hat{\mathbf{x}}_i, \mathbf{x}_k), LCS_{\delta,\gamma}(\mathbf{x}_i, \hat{\mathbf{x}}_k)\} \;\; otherwise \end{cases}, \tag{4}$$

where the parameter δ controls the window size that limits the temporal warping and $\gamma \in (0,1)$ is a constant threshold regulating the flexibility of the matching in space. According to the LCS similarity model, the distance between $\mathbf{x}_i$ and $\mathbf{x}_k$ can therefore be defined as

$$LCSD_{\delta,\gamma}(\mathbf{x}_i, \mathbf{x}_k) = 1 - \frac{LCS_{\delta,\gamma}(\mathbf{x}_i, \mathbf{x}_k)}{\min(n, s)}. \tag{5}$$

Like dynamic time warping, the distance based on LCS has a computational complexity of $O(\delta(n+s))$ and can be applied to time series of different length. However, it appeared to be less sensitive in the presence of outliers since it automatically stretches two sequences without compelling all the the points to be matched.

4 Experiments

The effectiveness of dbt-Isomap was assessed by means of computational tests concerning the classification of two time series gene expression data sets. Our aim was to determine whether dbt-Isomap may be profitably used for dimensionality reduction when it is combined with classical supervised learning algorithms,

represented by support vector machines (SVM) and the 1-nearest neighbor classifier (1NN). Moreover, we were interested in investigating whether distance measures other than standard Euclidean metric may represent valuable alternative to grasp gene expression similarity and build more effective data embeddings.

The first data set, denoted as *Yeast*, was originally analyzed in [25]. It contains the genome characterization of the mRNA transcript levels in nearly two full cell cycles of the yeast Saccharomyces cerevisiae. Gene expression levels were gathered at 17 time points, with an interval of ten minutes between each pair of recorded values. This data set consists of 388 time series associated to five different phases, namely Early G1 (67), Late G1 (136), S (77), G2 (54) and M (54), which represent the class labels in our experiments. The second data set, indicated as *MS-rIFN*β, was first proposed in [3]. It collects 52 gene expression profiles of patients suffering from relapsing-remitting multiple sclerosis, who are classified as either good (33) or poor responders (19) to recombinant human interferon beta (rIFNβ). In the original data set patients' profiles are composed by the expression levels of 70 genes isolated at 7 time points: before the administration of the drug ($t = 0$), every three months ($t = 1,2,3,4$) and every six months ($t = 5,6$) during the first and the second year of therapy. From the complete *MS-rIFN*β data set we retained only twelve genes whose expression profiles at $t = 0$ have shown to accurately predict the response to rIFNβ, as described in [3], and we generated six different data sets by extracting the gene expression series corresponding to each number of time points in the interval $[2, 7]$. Expression levels in *MS-rIFN*β were standardized before classification; each profile of $Yeast$ was instead normalized as described in [25]. Moreover, for each of the six *MS-rIFN*β data sets we sequenced genes and time periods for every patient, so to obtain a flat representation of the temporal sequences. Therefore, a given *MS-rIFN*β data set included $12T$ features, where T is the number of time points considered for each gene.

The comparison was performed by using either SVM or 1NN for classification. In particular, several competing algorithms were considered: the first two were represented by linear SVM and 1NN straightly applied to each data set in the original input space. The other methods, marked with the subscripts $I_{\mathcal{L}_1}$, $I_{\mathcal{L}_2}$, $I_{\mathcal{L}_\infty}$, I_{DTW} and I_{LCSD}, were obtained by resorting to dbt-Isomap alternatively combined with the distance metrics described in section 3; in this case, SVM and 1NN were used to classify the projection of each data set in the d-dimensional embedding space.

To automatically determine the value of d we considered the curve of residual variance obtained by dbt-Isomap, and searched for the dimension at which the curve ceased to decrease significantly, according to [6]. The intrinsic dimensionality d evaluated for each distance measure is reported in Table 1. Estimates were computed for dbt-Isomap by setting $k = 1/10m$ and $\varepsilon = 2/3d_{\max}$, where $d_{\max}$ is the maximum geodesic distance among all pairs of points in every data set.

The most promising parameters of each method were empirically found by minimizing the classification error rate within a preliminary 5-fold cross-validation

Table 1. Estimated dimensionality d of the projection space for dbt-Isomap with the alternative distance functions

Data set	$I_{\mathcal{L}_1}$	$I_{\mathcal{L}_2}$	$I_{\mathcal{L}_\infty}$	I_{DTW}	I_{LCSD}
Yeast	5	4	6	5	6
MS-rIFNβ					
t∈[0,1]	7	7	6	6	6
t∈[0,2]	8	8	7	6	8
t∈[0,3]	7	7	8	8	8
t∈[0,4]	8	8	8	8	9
t∈[0,5]	9	9	9	5	8
t∈[0,6]	9	9	9	6	8

run. In order to limit the grid search for dbt-Isomap, the radius of the hypersphere around each point was fixed to $\varepsilon = 2/3 d_{\max}$ for each data set and the number k of nearest neighbors was varied in the interval $[2, 1/10m]$ with a step size of 2. For DTW and LCSD, both the constrained and the warping-free variants were considered. In the first case, experiments were performed by setting the warping constant δ up to 30% of the time series length. For LCSD the matching coefficient γ took values in the interval $[0.2\sigma, \sigma]$, where σ denotes the standard deviation of the data set. Finally, the cost coefficient of linear SVM was set to $C = 10^j, j \in [-1, 3]$.

The classification accuracy was evaluated by means of ten times stratified 5-fold cross-validation by using the best parameters identified in the exploratory run. To guarantee a fair comparison the same folds for training and testing were used for all methods. Furthermore, the multi-category classification of *Yeast* by linear SVM was performed according to the one-against-all framework, in which a set of binary problems is obtained by discriminating between the points of one class and all the remaining. This scheme was not applied for 1NN which is able to address directly the multi-category nature of the classification task.

The average accuracy of ten times 5-fold cross-validation obtained by using SVM and 1NN as base learners is reported in tables 2 and 3, respectively. To support on a statistical basis these results we further applied the Wilcoxon signed-rank test to reject the null hypothesis that two classifiers performed equally well on the entire collection of data sets. In particular, the null hypothesis was rejected when the Wilcoxon statistic was greater or equal than the critical value for a two-tailed test with a significance level of 0.05. The outcome of this test is reported in tables 4 and 5, where each entry indicates whether the method on the row overall performed better (b), worse (w) or was equivalent (e) compared to the other.

The results presented in tables 2 and 3 suggest some empirical conclusions. Resorting to dimensionality reduction by means of dbt-Isomap often induced a significant improvement in accuracy with respect to the base case represented by linear SVM or 1NN applied in the original feature space. This remark held true independently of the data set and of the base learner. However, the performance

Table 2. Classification accuracy of ten times 5-fold cross-validation with linear SVM

	Method					
Data set	SVM	SVM$_{I_{\mathcal{L}_1}}$	SVM$_{I_{\mathcal{L}_2}}$	SVM$_{I_{\mathcal{L}_\infty}}$	SVM$_{I_{DTW}}$	SVM$_{I_{LCSD}}$
Yeast	0.753	0.751	**0.763**	0.754	0.561	0.744
MS-rIFNβ						
t∈[0,1]	0.775	0.914	**0.919**	0.901	0.848	0.854
t∈[0,2]	0.841	0.928	**0.938**	0.853	0.801	0.849
t∈[0,3]	0.838	**0.908**	0.885	0.848	0.819	0.864
t∈[0,4]	0.801	0.875	**0.915**	0.833	0.839	0.899
t∈[0,5]	0.786	**0.858**	0.849	0.812	0.839	0.838
t∈[0,6]	0.838	**0.868**	0.852	0.806	0.781	0.821

Table 3. Classification accuracy of ten times 5-fold cross-validation with 1NN

	Method					
Data set	1NN	1NN$_{I_{\mathcal{L}_1}}$	1NN$_{I_{\mathcal{L}_2}}$	1NN$_{I_{\mathcal{L}_\infty}}$	1NN$_{I_{DTW}}$	1NN$_{I_{LCSD}}$
Yeast	0.620	**0.700**	0.676	0.633	0.497	0.666
MS-rIFNβ						
t∈[0,1]	0.866	0.890	**0.916**	0.915	0.761	0.838
t∈[0,2]	0.828	**0.886**	0.878	0.881	0.764	0.828
t∈[0,3]	0.813	**0.923**	0.866	0.848	0.768	0.836
t∈[0,4]	0.725	0.864	**0.904**	0.867	0.789	0.809
t∈[0,5]	0.659	**0.844**	0.841	**0.844**	0.756	0.811
t∈[0,6]	0.746	**0.854**	0.826	0.787	0.760	0.746

showed by the competing algorithms drawn a clear distinction between methods based on lock-step measures, which consistently generated better classifications compared to direct SVM and 1NN, than those relying on elastic measures, for which improvements in accuracy were not always observed. In particular, classifiers based on $I_{\mathcal{L}_1}$ and $I_{\mathcal{L}_2}$ dominated those relying on I_{LCSD} and I_{DTW} in terms of rate of correct predictions. Regarding lock-step measures, better performances were achieved by using I_{LCSD} in place of I_{DTW} for dimensionality reduction.

Despite $I_{\mathcal{L}_1}$ and $I_{\mathcal{L}_2}$-based methods achieved the highest average accuracy on most data sets, they overall produced comparable results with respect to the algorithm relying on $I_{\mathcal{L}_\infty}$ when 1NN is used for classification, as indicated by the Wilcoxon test. To complement the previous experiment we therefore performed a second analysis by investigating the performance of the competing methods across all data sets. To this aim, for each algorithm we computed a score given by the ratio between its error rate and the lowest error rate on a given data set, both for SVM and 1NN-based classifiers. For a learning method the sum of the scores on all data sets can be regarded as a measure of its ability of generating

Table 4. Result of the Wilcoxon signed-rank test for SVM-based methods

	SVM	$SVM_{I_{\mathcal{L}_1}}$	$SVM_{I_{\mathcal{L}_2}}$	$SVM_{I_{\mathcal{L}_\infty}}$	$SVM_{I_{DTW}}$	$SVM_{I_{LCSD}}$
SVM	-	w	w	e	e	e
$SVM_{I_{\mathcal{L}_1}}$	b	-	e	b	b	b
$SVM_{I_{\mathcal{L}_2}}$	b	e	-	b	b	b
$SVM_{I_{\mathcal{L}_\infty}}$	e	w	w	-	e	e
$SVM_{I_{DTW}}$	e	w	w	e	-	w
$SVM_{I_{LCSD}}$	e	w	w	e	b	-

Table 5. Result of the Wilcoxon signed-rank test for 1NN-based methods

	1NN	$1NN_{I_{\mathcal{L}_1}}$	$1NN_{I_{\mathcal{L}_2}}$	$1NN_{I_{\mathcal{L}_\infty}}$	$1NN_{I_{DTW}}$	$1NN_{I_{LCSD}}$
1NN	-	w	w	w	e	e
$1NN_{I_{\mathcal{L}_1}}$	b	-	e	e	b	b
$1NN_{I_{\mathcal{L}_2}}$	b	e	-	e	b	b
$1NN_{I_{\mathcal{L}_\infty}}$	b	e	e	-	b	b
$1NN_{I_{DTW}}$	e	w	w	w	-	w
$1NN_{I_{LCSD}}$	e	w	w	w	b	-

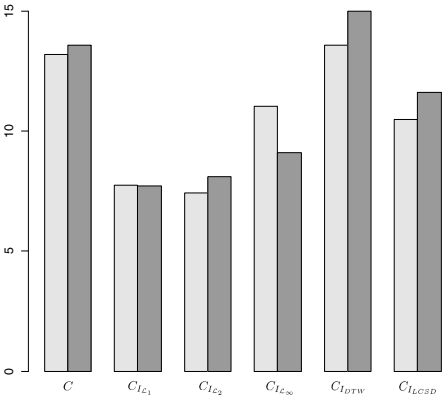


Fig. 1. Sum of the scores over all data sets for each method. Light grey bars refer to SVM classifiers; dark grey bars to 1NN methods. The first two columns on the left represent the score of the base classifier C directly applied in the original input space. The other columns indicate the scores obtained by combining C with dbt-Isomap with the alternative distance functions.

classifications which are close, if not equal, to the best ones in terms of accuracy. The lower the score is, the better the method performed. The results of this study, depicted in figure 1, further highlighted the effectiveness of the classifiers based

on $I_{\mathcal{L}_1}$ and $I_{\mathcal{L}_2}$, which achieved the two lowest total score being associated with ratios more densely distributed around 1. According to this analysis, methods based on dbt-Isomap with $\mathcal{L}_\infty$-norm confirmed themselves as the third best alternative for classification.

5 Conclusions and Future Extensions

The paper presents an empirical evaluation of double-bounded tree connected Isomap (dbt-Isomap) for nonlinear dimensionality reduction in the context of time series gene expression data classification. In order to cope with the embedding of temporal sequences, dbt-Isomap is combined with alternative distance functions which have been widely proposed as powerful time series similarity measures. Computational experiments were performed on two benchmark time series gene expression data sets. They showed the usefulness of resorting to dbt-Isomap for dimensionality reduction, especially when lock-step measures are used for building the neighborhood graph. More specifically, they highlighted the effectiveness of $\mathcal{L}_1$-norm which can be regarded as the best alternative to the Euclidean metric for projecting time series gene expression data.

The present study suggests several directions for future research. First, the usefulness of different manifold learning methods which have proven to be rather effective for microarray data classification, such as locally linear embedding and Laplacian eigenmaps, should be investigated. Furthermore, it would be worthwhile to analyze for the same classification task the performance of other Isomap variants, represented by supervised or kernelized extensions.

References

1. Peddada, S., Lobenhofer, E., Li, L., Afshari, C., Weinberg, C., Umbach, D.: Gene selection and clustering for time-course and dose-response microarray experiments using order-restricted inference. Bioinformatics 19, 834–841 (2003)
2. Hamadeh, H., Bushel, P., Paules, R., Afshari, C.: Discovery in toxicology: mediation by gene expression array technology. Journal of Biochemical and Molecular Toxicology 15, 231–242 (2001)
3. Baranzini, S., Mousavi, P., Rio, J., Caillier, S., Stillman, A., Villoslada, P., Wyatt, M., Comabella, M., Greller, L., Somogyi, R., Montalban, X., Oksenberg, J.: Transcription-based prediction of response to IFNβ using supervised computational methods. PLoS Biology 3, 166–176 (2005)
4. Jolliffe, I.T.: Principal component analysis. Springer, New York (1986)
5. Cox, T.F., Cox, M.A.A.: Multidimensional Scaling. Chapman and Hall, London (1994)
6. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290, 2319–2323 (2000)
7. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science 290, 2323–2326 (2000)
8. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation 15, 1373–1396 (2003)

9. Orsenigo, C., Vercellis, C.: A comparative study of nonlinear manifold learning methods for cancer microarray data classification. Expert Systems with Applications 40, 2189–2197 (2013)
10. Mizuhara, Y., Hayashi, A., Suematsu, N.: Embedding of Time Series Data by Using Dynamic Time Warping Distances. Systems and Computers in Japan 37, 241–249 (2006)
11. Balasubramanian, M., Schwartz, E.L., Tenenbaum, J.B., de Silva, V., Langford, J.C.: The Isomap algorithm and topological stability. Science 295, 7 (2002)
12. Orsenigo, C., Vercellis, C.: An effective double-bounded tree-connected Isomap algorithm for microarray data classification. Pattern Recognition Letters 33, 9–16 (2012)
13. Dawson, K., Rodriguez, R.L., Malyj, W.: Sample phenotype clusters in high-density oligonucleotide microarray data sets are revealed using Isomap, a nonlinear algorithm. BMC Bioinformatics 6, 195 (2005)
14. Park, H.: ISOMAP induced manifold embedding and its application to Alzheimer's disease and mild cognitive impairment. Neuroscience Letters 513, 141–145 (2012)
15. Geng, X., Zhan, D.C., Zhou, Z.H.: Supervised nonlinear dimensionality reduction for visualization and classification. IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics 35, 1098–1107 (2005)
16. Orsenigo, C., Vercellis, C.: Kernel ridge regression for out-of-sample mapping in supervised manifold learning. Expert Systems with Applications 39, 7757–7762 (2012)
17. Wang, X., Mueen, A., Ding, H., Trajcevski, G., Scheuermann, P., Keogh, E.: Experimental comparison of representation methods and distance measures for time series data. Data Mining and Knowledge Discovery (2012), doi:10.1007/s10618-012-0250-5
18. Yi, B.-K., Faloutsos, C.: Fast time sequence indexing for arbitrary Lp norms. In: Proc. of the 26th International Conference on Very Large Data Bases, pp. 385–394. Morgan Kaufmann, San Francisco (2000)
19. Sakoe, H., Chiba, C.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustics, Speech and Signal Processing 26, 43–49 (1978)
20. Keogh, E., Ratanamahatana, C.A.: Exact indexing of dynamic time warping. Knowledge and Information Systems 7, 358–386 (2004)
21. Vlachos, M., Hadjieleftheriou, M., Gunopulos, D., Keogh, E.: Indexing multidimensional time-series. The VLDB Journal 15, 1–20 (2006)
22. Orsenigo, C., Vercellis, C.: Combining discrete SVM and fixed cardinality warping distances for multivariate time series classification. Pattern Recognition 43, 3787–3794 (2010)
23. Keogh, E.J., Ratanamahatana, C.A.: Exact indexing of dynamic time warping. Knowledge and Information Systems 7, 358–386 (2005)
24. Vlachos, M., Gunopulos, D., Kollios, G.: Discovering similar multidimensional trajectories. In: Proc. of the 18th International Conference on Data Engineering, pp. 673–684. IEEE Computer Society, Washington (2002)
25. Cho, R.J., Campbell, M.J., Winzeler, E.A., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T.G., Gabrielian, A.E., Landsman, D., Lockhart, D.J., Davis, R.W.: A genome-wide transcriptional analysis of the mitotic cell cycle. Molecular Cell 2, 65–73 (1998)

Supervising Random Forest Using Attribute Interaction Networks

Qinxin Pan[1], Ting Hu[1], James D. Malley[4], Angeline S. Andrew[2,3], Margaret R. Karagas[2,3], and Jason H. Moore[1,2,3]

[1]Department of Genetics, Geisel School of Medicine, Dartmouth College, Hanover, NH 03755, USA
[2]Department of Community and Family Medicine, Geisel School of Medicine, Dartmouth College, Hanover, NH 03755, USA
[3]Institute for Quantitative Biomedical Sciences, Dartmouth College, Hanover, NH 03755, USA
[4]Division of Computational Bioscience, Center for Information Technology, National Institutes of Health, Bethesda, MD 20892, USA
jason.h.moore@dartmouth.edu

Abstract. Genome-wide association studies (GWAS) have become a powerful and affordable tool to study the genetic variations associated with common human diseases. However, only few of the loci found are associated with a moderate or large increase in disease risk and therefore using GWAS findings to study the underlying biological mechanisms remains a challenge. One possible cause for the "missing heritability" is the gene-gene interactions or epistasis. Several methods have been developed and among them Random Forest (RF) is a popular one. RF has been successfully applied in many studies. However, it is also known to rely on marginal main effects. Meanwhile, networks have become a popular approach for characterizing the space of pairwise interactions systematically, which can be informative for classification problems. In this study, we compared the findings of Mutual Information Network (MIN) to that of RF and observed that the variables identified by the two methods overlap with differences. To integrate advantages of MIN into RF, we proposed a hybrid algorithm, MIN-guided RF (MINGRF), which overlays the neighborhood structure of MIN onto the growth of trees. After comparing MINGRF to the standard RF on a bladder cancer dataset, we conclude that MINGRF produces trees with a better accuracy at a smaller computational cost.

Keywords: Random Forest, Mutual Information Network, Mutual Information Network guided Random Forest, Classification.

1 Introduction

The current strategy for studying the genetic basis of disease susceptibility is to measure millions of single nucleotide polymorphisms (SNPs) across the human genome and test each of them individually for association [14,30]. Genome-wide

L. Vanneschi, W.S. Bush, and M. Giacobini (Eds.): EvoBIO 2013, LNCS 7833, pp. 104–116, 2013.

association studies (GWAS) are based on the idea that genetic variations with alleles common in the population will additively explain much of the heritability of common diseases. As the cost for genome-wide genotyping decreases, the number of GWAS has increased considerably and this approach is now relatively common. The GWAS approach has been successful in that hundreds of new disease-associated SNPs have been reported using rigorous statistical significance and replication criteria [20]. It is anticipated that those SNPs will reveal new pathobiology that will in turn lead to new treatments. While this may be true, few of the loci identified are associated with a moderate or large increase in disease risk and some empirically identified genetic risk factors have been missed [25]. At best, about 20% of the total genetic variance has been explained for a few select common diseases such as the Crohns disease [12]. As a result, many have asked where the missing heritability is [11]. One possibility is that complexities such as gene-gene interactions or epistasis can limit the power of analysis approaches that only consider one SNP at a time [22,23,31].

To faithfully capture the relationships among SNPs several machine learning methods have been considered, including Random Forest (RF) [8,21]. These are, however, engines for making predictions and not necessarily for declaring complexity among the features in the prediction. RF, in particular, is driven by estimating marginal effects in its tree-building process and this is not necessarily a complexity-seeking scheme. RF is one of the most popular ensemble learning methods and has many applications [4,6,10]. A decision tree classifies subjects as case or control by sorting them through a tree from node to node, where each node is a variable with a decision rule that guides that subject through different branches of the tree to a leaf that provides its classification [3]. A RF is a collection of individual decision tree classifiers, where each tree in the forest is trained using a bootstrap sampling of instances (i.e. subjects) from the data, and each variable in the tree is chosen from a random subset of variables. Classification of instances is based upon aggregate voting over all trees in the forest [3]. Although powerful, decision-tree-based methods have one major limitation, that the standard implementations condition on marginal effects [28]. In other words, the algorithm finds the best single variable for the root node before adding additional variables as nodes to the model. This can preclude the detection of epistasis in the absence of significant single SNP effects [27,28,31]. Moreover, multiple variables randomly drawn from the dataset are evaluated and only the best one among them is used for subject sorting. The evaluation of multiple random variables makes RF computationally expensive.

Meanwhile, network science has been used to model interactions and dependencies [1,7,16]. Recently, Hu *et al* [15] proposed Statistical Epistasis Networks (SEN) to characterize the space of pairwise interactions in population-based genetic association studies [18]. In the network, each vertex corresponds to a SNP. An edge linking a pair of vertices corresponds to an interaction between two SNPs. Weights assigned to each SNP and each pair of SNPs quantify how much of the disease status the corresponding SNP and SNPs pair can explain. SEN displays a global representation of all pairwise neighborhood relationship, which

could potentially provide information to supervise classification. However, the network alone does not make a prediction, which makes it hard to interpret.

Given the strengths and weaknesses of RF and networks, we consider a hybrid scheme that has the strengths of both and the weaknesses of neither. Specifically, we embed the unsupervised process of network building into the supervised process of prediction: we use the structure of networks to guide the growing of the forest. In this way, we are able to use the knowledge about variable-variable relationship during the growth of trees, which could potentially make RF less biased towards the marginal main effects, and more efficient by avoiding a random sampling of variables.

The above approach, named Mutual Information Network guided Random Forest (MINGRF), is compared with standard RF on a population-based bladder cancer dataset. The results show that MINGRF produces trees with better accuracies in a shorter runtime.

2 Methods

2.1 Bladder Cancer Dataset

The dataset used in this study consisted of cases of bladder cancer among New Hampshire residents, 25 to 74 years of age, diagnosed from July 1, 1994 to December 31, 2001, and identified in the State Cancer Registry. Controls less than 65 years of age were selected using population lists obtained from the New Hampshire Department of Transportation. Controls 65 years of age and older were chosen from data files provided by the Centers for Medicare & Medicaid Services (CMS) of New Hampshire. This dataset shared a control group with a study of non-melanoma skin cancer in New Hampshire covering an overlapping diagnostic period of July 1, 1993 to June 30, 1995. Additional controls for bladder cancer cases diagnosed from July 1, 1995 to June 30, 1998 were selected with matching age and gender.

Genotyping was performed using the GoldenGate Assay System. The missing value of an individual was filled using the most common genotype of corresponding SNP in the population. The dataset used in our analysis consisted of 491 bladder cancer cases and 71 controls. 1,422 SNPs are included in the dataset. More details on this dataset and the methods are available in [2,17].

2.2 Mutual Information Network

In mathematical terms, a network is a graph, where a graph G consists of a set $V(G)$ of vertices and a set $E(G)$ of edges [24]. In our Mutual Information Networks (MIN), each vertex corresponds to a SNP, and we use v_{A} to denote the vertex corresponding to SNP A. An edge linking a pair of vertices, for instance v_{A} and v_{B}, represents an interaction between SNPs A and B. We first assigned a weight to each pair of SNPs to quantify how much of the disease status the corresponding SNP pair genotypes together explain. In information theoretic terms, the weight corresponds to the two-way *mutual information* [9]. Specifically, the

weight of the edge connecting v_{A} and v_{B} is $I(A, B; C)$, the mutual information of SNPs A and B together with C, the class variable with status *case* or *control*. Intuitively, $I(A, B; C)$ is the reduction in the uncertainty of the class C due to knowledge about SNP A and B's genotypes. Its precise definition is

$$I(A, B; C) = H(C) - H(C|A, B), \tag{1}$$

where $H(C)$ is the *entropy* of C, i.e., the measure of the uncertainty of class C, and $H(C|A, B)$ is the *conditional entropy* of C given knowledge of SNP A and B. Entropy and conditional entropy are defined by

$$H(C) = \sum_{c} p(c) \log \frac{1}{p(c)}, \tag{2}$$

$$H(C|A, B) = \sum_{a,b,c} p(a, b, c) \log \frac{1}{p(c|a, b)}, \tag{3}$$

where $p(c)$ is the probability that an individual has class c, $p(a, b, c)$ is that of having genotype a, b and class c, and $p(c|a, b)$ is that of having class c given the occurrence of genotype a and b together.

Similar to the framework of Statistical Epistasis Network by Hu *et al*, the threshold of including pairwise interactions can be derived systematically by analyzing the topological properties of the networks [15], such as the size of a network, the connectivity of a network (the size of its largest connected component), and its vertex degree distribution. Permutation testing is often used to provide a null distribution of properties of networks built from permuted data. This null distribution can be used to determine the threshold of pairwise strength that mostly distinguishes the real-data network from the permuted-data networks.

2.3 Random Forest

Random Forest is an ensemble learning method. A forest consists of multiple decision tree classifiers and the classification of subjects is based on aggregate voting over all trees.

Specifically, a standard RF procedure takes the following steps [3,6,19]: i) draw *ntree* bootstrap samples from the original data; ii) grow a tree for each bootstrap dataset. At each node of the tree, randomly select *mtry* variables and choose the splitting node that separates cases and controls the best. iii) a tree grows to the largest extent when the number of subjects in a node reaches a minimum *nodesize* and the prediction of that node is decided by the majority class of subjects on that node; iv) aggregate information from the *ntree* trees for new data prediction such as majority voting for classification; v) compute an out-of-bag (oob) accuracy by using the data not in the bootstrap sample [5]. The balanced accuracy, that is the average of sensitivity and specificity, is reported in this study as it is more robust to imbalanced biomedical datasets [29].

2.4 Mutual Information Network Guided Random Forest

We investigate whether MIN can help improve RF by implementing a hybrid algorithm, Mutual Information Network guided Random Forest (MINGRF). To impose the structure of MIN into RF, we implement MINGRF in the following way.

1. When starting building trees, instead of sampling a random set of variables and choosing the one which separates cases and controls best, MINGRF chooses one vertex from the hubs in MIN (vertices that have at least 5 neighbors specified in this study) with a probability proportional to their degrees.
2. While growing the trees, instead of trying a list of variables and choosing the best-case-control-separating one, MINGRF considers all the neighbors of the mother node in MIN which have not been used yet in the building of the current tree and chooses one with probability proportional to the corresponding edge weights. If all neighbors have been used previously, MINGRF chooses one of them with a probability proportional to the edge weights.
3. The growing of a tree continues until the number of samples on a node is smaller than a pre-specified number, i.e. the terminal node size.
4. After the construction of the forest, the quality of trees can be assessed using oob samples in the same manner as standard RF.

In this study, to compare the performances of MINGRF and RF, we use $ntree = 1000$. A wide range of $mtry$ and $nodesize$ are explored in our implementation.

3 Results

3.1 Mutual Information Network

To pick the threshold t at which mutual information network is most different from by chance, we look into the connectivity of the network, i.e. the size of the largest connected component, at decreasing t. Recall that an edge linking SNPs A and B is included in the mutual information network G_t only if their two-way mutual information $I(A, B; C) \geq t$. Accordingly, the networks G_t grow as t decreases.

Figure 1 shows the size of the largest connected component in the network G_t and in the permuted-data networks as t decreases from 0.030 to 0.005 in increments of 0.0001. The largest connected component of G_t grows quickly when t decreases from 0.025 to 0.0156 whereas the largest connected component in the permuted-data networks do not start growing until a smaller value of t is reached. The P value of the largest connected component size, estimated based on permutation testing, is smaller than 0.001 when $t \in [0.0155, 0.0299]$. We choose $G_{0.0156}$ for future study as all 1,422 SNPs were included in the largest connected component for the first time when t reaches 0.0156.

The network $G_{0.0156}$ (Figure 2) has 1,422 vertices and 2,236 edges. As SNP IGF2AS_04 has the strongest main effect, every SNP pair which includes IGF2AS

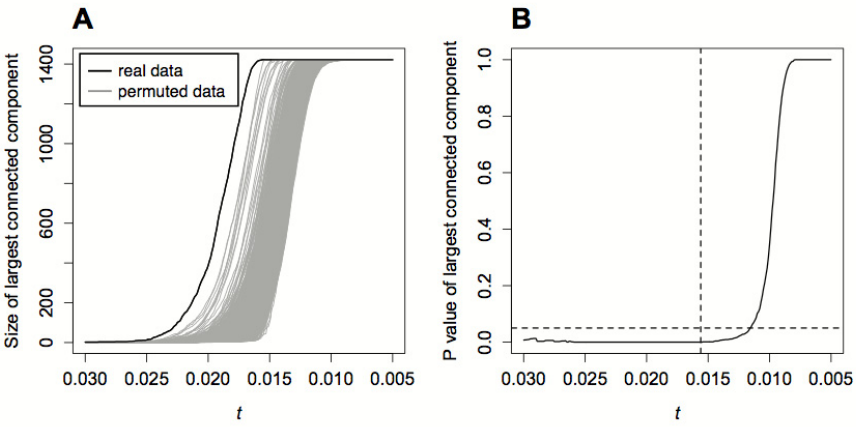


Fig. 1. The size of the largest connected component and its significance in the networks with decreasing threshold t. (A) The size of the largest connected component in G_t and networks of permuted datasets. The black line represents the real-data network G_t and the gray lines represent the networks of 1,000 permuted datasets. The largest connected components include increasingly more vertices as t decreases and eventually include all 1,422 vertices when t reaches 0.0156. (B) The P value of the largest connected component in G_t. P value is estimated as the fraction of networks from permuted datasets whose largest connected component is no smaller than that of G_t. The horizontal dashed line represents P=0.05 and the vertical dashed line represents t=0.0156.

_04 has a relatively high $I(A, B; C)$. Naturally, vertex IGF2AS_04 is connected to every other vertex in $G_{0.0156}$ and has a degree of 1,421. There are 861 vertices which are only connected to IGF2AS_04 (shown on the left) and 560 vertices which have at least one more neighbor besides IGF2AS_04 (shown on the right). Note that there are a large set of edges in which IGF2AS_04 is not involved. In other words, interactions or additive effects that are independent of IGF2AS_04 also contribute to bladder cancer risk, which further indicates the fact that bladder cancer is a complex disease.

3.2 Mutual Information Network and Random Forest Network Comparison

To compare MIN and RF, we ask the research question whether important variables in RF are also identified as important in MIN. Gini importance, which indicates both how often a particular variable is selected for a split and how large its overall discriminative value is for the classification problem under study, is used to quantify the importance of a particular variable in RF. Node degree, defined as the number of neighbors a specific vertex has in the graph, is used to assess the importance of that variable in MIN. Recall that MIN $G_{0.0156}$ is chosen as all 1,422 SNPs are included in the largest connected component when

Fig. 2. Mutual Information Network $G_{0.0156}$. There are 1,422 vertices and 2,236 edges. As vertex IGF2AS_04 (shown in the middle) has strong main effect which contributes to two-way mutual information, it is connected to every other vertex in the graph. For visualization purpose, all edges that connect vertex IGF2AS_04 are shown in gray all the other edges are shown in black. Vertices which are only connected to IGF2AS_04 are shown on the left side and vertices which have at least one more other neighbors besides IGF2AS_04 are shown on the right side. The graph is generated by the software Cytoscape [26].

the threshold $t \leq 0.0156$ (Figure 1). Figure 4A shows a significant correlation between the degree of MIN $G_{0.0156}$ and RF Gini importance (Spearman's rank correlation coefficient ρ=0.495 and $P<2.2 \times 10^{-16}$).

As RF Gini importance is known to be biased towards main effect [27,28], to fully compare MIN and RF with SNP-SNP relationships taken into account, we convert a forest into a RF network and compared it with MIN. Given a forest, we count the occurrence of two SNPs being mother-daughter nodes in all the trees and assign the occurrence as the weight to the edge between the SNP pair. Similar to the threshold-based MIN, an edge is included in the RF network $\hat{G}_t$ only when its weight is no less than a particular threshold t. Figure 3 shows the number of edges and the size of largest connected component in the network of real data and networks of permuted data as t decreases from 4 to 1 in increments of 1. When $t \leq 2$, the networks of permuted data possessed much more edges than that of real data, whereas when $t \geq 3$ the difference became negligible (Figure 3A). The size of largest connected component in networks of permuted data was significantly larger than that of real data when t=2 (Figure 3B). Based on above observations, we chose t=2 as a threshold for later study.

After obtaining a RF network $\hat{G}_2$, we are able to compare its node degree with that of the MIN $G_{0.0156}$. A significant correlation between the degree of MIN $G_{0.0156}$ and that of RF network G_2 is observed with Spearman's rank correlation coefficient ρ=0.483 and $P< 2.2 \times 10^{-16}$ (Figure 4B). Although correlated, the low correlation coefficients indicate the fact that the methods overlap with differences.

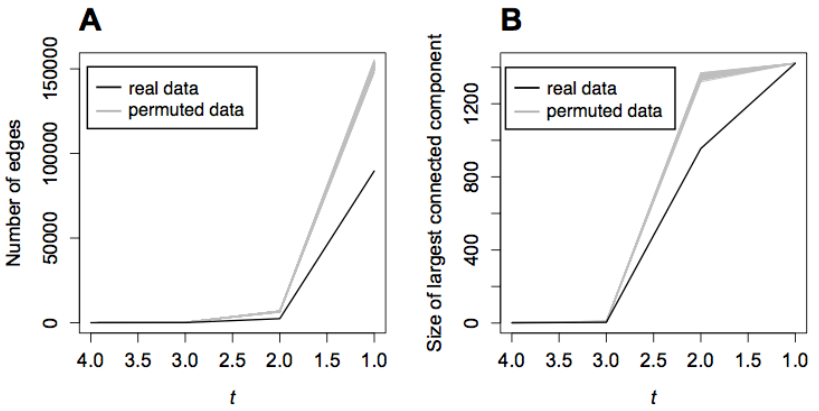


Fig. 3. Random Forest Network growth with decreasing threshold t. (A) Increase in the number of edges. (B) Increase in the size of largest connected component. In both panels, the black line represents $\hat{G}t$ of the real data and the gray lines represent networks of 1,00 permuted datasets. The threshold t, denoted as the times two corresponding SNPs show up in the forest as mother-daughter nodes, decreases from 4 to 1 in increments of 1.

3.3 Mutual Information Network Guided Random Forest

We evaluate the performance of MIN guided RF (MINGRF) using out-of-bag (oob) accuracy and runtime (Figure 5). Recall that RF has three key parameters: the terminal node size, the number of variables randomly sampled at each splitting and the number of trees in the forest. As MINGRF does not randomly sample a set of variables at each splitting, it has only two: the terminal node size and the number of trees. We thoroughly compare their performances under a wide range of different parameters and find that MINGRF always has better oob accuracy than RF (Figure 5A and 5B). Although RF oob accuracy increases as the number of variables sampled increases, the runtime of RF also increases accordingly (Figure 5B and 5C). The runtime of RF is shorter than that of MINGRF when the number of variables sampled is small, but it exceeds that of MINGRF quickly when the number of variables sampled starts to increase (Figure 5C).

4 Discussion

In this article, we compare the findings of Mutual Information Network and Random Forest in a bladder cancer dataset and observe both similarities and differences. The differences allow the potential for improvement which might be achieved by combining the two methods. Encouraged by these findings, to further integrate the advantages of MIN into RF, we propose a hybrid algorithm by imposing the neighborhood relationship of MIN into the tree construction of RF, i.e. MIN guided RF (MINGRF). Usually, RF randomly samples a list of variables

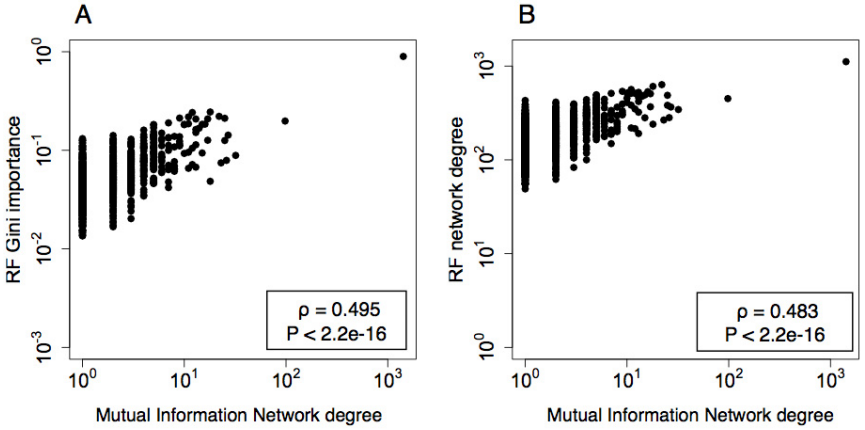


Fig. 4. Comparison of Mutual Information Network $G_{0.0156}$ and Random Forest. (A) Correlation of Mutual Information Network $G_{0.0156}$ node degree and RF Gini importance. (B) Correlation of Mutual Information Network $G_{0.0156}$ node degree and RF Network G_2 node degree. In both panels, Spearman's rank correlation coefficient ρ and the corresponding P value are reported.

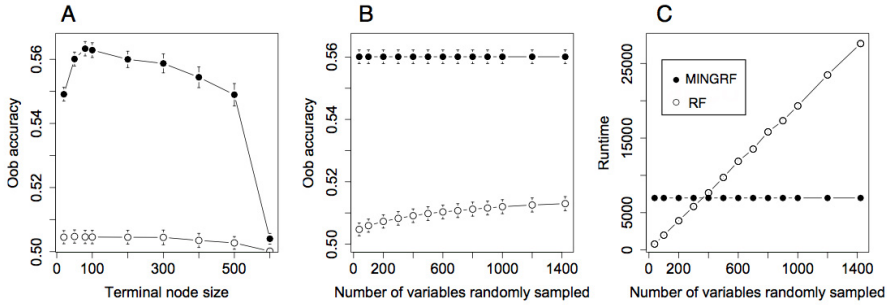


Fig. 5. Performance of Mutual Information Network Guided RF and RF. (A) Balanced oob accuracy shown as a function of terminal node size. (B) Balanced oob accuracy shown as a function of number of variables randomly sampled at each splitting. (C) Run time shown as a function of number of variables randomly sampled at each splitting. Data represent the mean of 1,000 independent replications and error-bars denote 95% confidence intervals.

and greedily chooses the one which separates cases and controls the best at each splitting. This process relies on the marginal main effect and is computationally expensive. In contrast, MINGRF takes advantage of the pairwise interaction landscape and put the strong pairs in MIN adjacent to each other in the tree, which not only takes SNP-SNP relationship (i.e. interaction, additive effect) into account but also improves computational efficiency. We find that the trees

produced by MINGRF have better oob accuracies and shorter runtime for a broad range of different parameters.

The comparison of MIN and RF leads to a few interesting observations. First, the MIN of real data has more vertices on the largest connected component than permuted data (Figure 1). As the edge number of real data and that of permuted data are not significantly different (data not shown), the difference of the largest connected component size is most likely caused by the clustering of interacting SNPs, in other words, the existence of hubs. Second, the RF networks of permuted datasets possess more edges and more vertices on the largest connected components (Figure 3). This could be partially explained by the fact that in a permuted dataset without real biological signals, it is hard to separate cases and controls and consequently RF learns random noises. Thus, it takes more steps of splitting to reach a certain terminal node size in a permuted dataset than in the real dataset, which leads to more edges in the permuted-data networks. The clear discrepancy between RF network of real data and that of permuted data indicates that our approach of constructing RF network captures the characteristic of the dataset and thus the RF network is comparable with MIN. Third, the positive correlation observed between MIN and RF indicate that the two methods identify similar variables. However, given the low correlation coefficients, their difference is definitely not negligible (Figure 4). There are a few possible reasons for the difference: i) MIN is deterministic whereas the sampling process in RF can introduce stochastic noises. Therefore, for variables with low MIN degree, their RF Gini importances can vary a lot; ii) MIN considers SNP-SNP relationships while RF mostly relies on marginal main-effect. SNP pairs which are in interaction without strong marginal main effects would be captured by MIN but not necessarily by RF.

Based on the above observations, we implement a hybrid algorithm called Mutual Information Network guided Random Forest (MINGRF). The goal of designing this algorithm is to refine the random sampling process of RF and consequently improve both the tree quality and runtime. MINGRF has many advantages. i) The choice of significant MIN does not rely on the significance of each pair of SNPs, instead it describes the point when the network as a whole system is most significant; ii) MINGRF takes advantage of the relationship between two variables and is less biased towards main effects; iii) MINGRF produces trees with better accuracies, which could be informative to propose a biological hypothesis; iv) MINGRF avoids the random sampling of variables at each splitting, which makes the construction of trees more efficient.

Among the limitations of this approach, an important one is that as MIN only captures pairwise SNP-SNP relationships, and higher order interactions might be overlooked. Moreover, with the growth of trees in the forest, RF identifies variables which separate cases and controls in a small subset of samples falling on the corresponding node, which encourages the detection of heterogeneity. But MINGRF finds variables essential for the whole population due to the way we construct MIN, which makes it not very useful for heterogeneity. As an alternative we could consider finding interacting features within the subpopulations,

to build networks that are possibly group dependent. We could then test for differences between the two sets of detected networks. There are simple matrix methods for this. If the group-based networks are declared similar it makes sense to declare them as valid population networks. Otherwise group differences would be captured by the separate networks, and these would be adapted to heterogeneity.

Future work includes comparing MINGRF with RF more thoroughly using cross fold validation, tree consistency etc. As the construction process is more transparent in MINGRF, we expect to get more interpretable models. Moreover, we are also interested in studying the top variables identified by MINGRF and RF. Whether the top variables are truly in interactions can be tested using explicit test [13]. Whether the usage of local neighborhood relationships in MIN will help the findings of higher order interactions will also be interesting to investigate. As MINGRF uses information about SNP-SNP relationships, we expect MINGRF to detect interactions which are usually overlooked by standard RF [28].

In conclusion, we compare two methods, Mutual Information Network (MIN) and Random Forest (RF), and observed both similarities and differences. MIN captures the two-way interaction landscape well yet can not give a prediction itself. On the other hand, RF is powerful in classification problems but also is known to be biased towards marginal main effects. After a thorough comparison, we propose a novel algorithm MIN guided RF (MINRF) and test it on a bladder cancer dataset. We conclude that MINRF yields decision trees with better accuracies at a lower computational cost.

Acknowledgments. This work was supported by NIH grants R01-LM009012, R01-LM010098, and R01-AI59694.

References

1. Andrei, A., Kendziorski, C.: An efficient method for identifying statistical interactors in gene association networks. Biostatistics 10(4), 706–718 (2009)
2. Andrew, A.S., Nelson, H.H., Kelsey, K.T., Moore, J.H., Meng, A.C., Casella, D.P., Tosteson, T.D., Schned, A.R., Karagas, M.R.: Concordance of multiple analytical approaches demonstrates a complex relationship between dna repair gene snps, smoking and bladder cancer susceptibility. Carcinogenesis 27(5), 1030–1037 (2006)
3. Breiman, L.: Random forests. Mach. Learn. 45, 5–32 (2001)
4. Bureau, A., Dupuis, J., Falls, K., Lunetta, K.L., Hayward, B., Keith, T.P., Van Eerdewegh, P.: Identifying snps predictive of phenotype using random forests. Genet. Epidemiol. 28(2), 171–182 (2005)
5. Bylander, T.: Estimating generalization error on two-class datasets using out-of-bag estimates. Machine Learning 48, 287–297 (2002)
6. Chen, X., Ishwaran, H.: Random forests for genomic data analysis. Genomics 99(6), 323–329 (2012)
7. Chu, J.H., Weiss, S.T., Carey, V.J., Raby, B.A.: A graphical model approach for inferring large-scale networks integrating gene expression and genetic polymorphism. BMC Syst. Biol. 3, 55 (2009)

8. Cordell, H.J.: Epistasis: what it means, what it doesn't mean, and statistical methods to detect it in humans. Hum. Mol. Genet. 11(20), 2463–2468 (2002)
9. Cover, T.M., Thomas, J.A.: Elements of information theory, 2nd edn. Wiley (2006)
10. Diaz-Uriarte, R., Alvarez de Andres, S.: Gene selection and classification of microarray data using random forest. BMC Bioinformatics 7, 3 (2006)
11. 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)
12. Franke, A., McGovern, D.P.B., Barrett, J.C., Wang, K., Radford-Smith, G.: Genome-wide meta-analysis increases to 71 the number of confirmed crohn's disease susceptibility loci. Nat. Genet. 42(12), 1118–1125 (2010)
13. Greene, C.S., Himmelstein, D.S., Nelson, H.H., Kelsey, K.T., Williams, S.M., Andrew, A.S., Karagas, M.R., Moore, J.H.: Enabling personal genomics with an explicit test of epistasis. In: Pac. Symp. Biocomput., pp. 327–336 (2010)
14. Hirschhorn, J.N., Daly, M.J.: Genome-wide association studies for common diseases and complex traits. Nat. Rev. Genet. 6(2), 95–108 (2005)
15. Hu, T., Sinnott-Armstrong, N.A., Kiralis, J.W., Andrew, A.S., Karagas, M.R., Moore, J.H.: Characterizing genetic interactions in human disease association studies using statistical epistasis networks. BMC Bioinformatics 12(364) (2011)
16. Ideker, T., Sharan, R.: Protein networks in disease. Genome Res. 18(4), 644–652 (2008)
17. Karagas, M.R., Tosteson, T.D., Blum, J., Morris, J.S., Baron, J.A., Klaue, B.: Design of an epidemiologic study of drinking water arsenic exposure and skin and bladdder cancer risk in a U.S. population. Environ. Health Perspect. 106(suppl. 4), 1047–1050 (1998)
18. Lavender, N.A., Rogers, E.N., Yeyeodu, S., Rudd, J., Hu, T., Zhang, J., Brock, G.N., Kimbro, K.S., Moore, J.H., Hein, D.W., Kidd, L.C.R.: Interaction among apoptosis-associated sequence variants and joint effects on aggressive prostate cancer. BMC Med. Genomics 5, 11 (2012)
19. Malley, J.D., Kruppa, J., Dasgupta, A., Malley, K.G., Ziegler, A.: Probability machines: Consistent probability estimation using nonparametric learning machines. Methods Inf. Med. 10(51), 74–81 (2011)
20. Manolio, T.A.: Genomewide association studies and assessment of risk of disease. New England Journal of Medicine 363(2), 166–176 (2010)
21. McKinney, B.A., Reif, D.M., Ritchie, M.D., Moore, J.H.: Machine learning for detecting gene-gene interactions: a review. Appl. Bioinformatics 5(2), 77–88 (2006)
22. Moore, J.H., Asselbergs, F.W., Williams, S.M.: Bioinformatics challenges for genome-wide association studies. Bioinformatics 26(4), 445–455 (2010)
23. Moore, J.H., Williams, S.M.: Epistasis and its implications for personal genetics. Am. J. Hum. Genet. 85(3), 309–320 (2009)
24. Newman, M.: Networks: An introduction. Oxford University Press (2010)
25. Ritchie, M.D., Hahn, L.W., Roodi, N., Bailey, L.R., Dupont, W.D., Parl, F.F., Moore, J.H.: Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer. Am. J. Hum. Genet. 69(1), 138–147 (2001)
26. Shannon, P., Markiel, A., Ozier, O., Baliga, N.S., Wang, J.T., Ramage, D., Amin, N., Schwikowski, B., Ideker, T.: Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res. 13(11), 2498–2504 (2003)

27. Strobl, C., Boulesteix, A.L., Kneib, T., Augustin, T., Zeileis, A.: Conditional variable importance for random forests. BMC Bioinformatics 9(307) (2008), doi:10.1186/1471–2105–9–307
28. Strobl, C., Boulesteix, A.L., Zeileis, A., Hothorn, T.: Bia in random forest variable importance measures: Illustration, sources and a solution. BMC Bioinformatics 8(25) (2007), doi:10.1186/1471–2105–8–25
29. 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(4), 306–315 (2007)
30. Wang, W.Y.S., Barratt, B.J., Clayton, D.G., Todd, J.A.: Genome-wide association studies: theoretical and practical concerns. Nat. Rev. Genet. 6(2), 109–118 (2005)
31. Williams, S.M., Canter, J.A., Crawford, D.C., Moore, J.H., Ritchie, M.D., Haines, J.L.: Problems with genome-wide association studies. Science 316(5833), 1840–1842 (2007)

Hybrid Genetic Algorithms for Stress Recognition in Reading

Nandita Sharma and Tom Gedeon

Research School of Computer Science
Australian National University, Canberra, Australia
{Nandita.Sharma,Tom.Gedeon}@anu.edu.au

Abstract. Stress is a major problem facing our world today and affects everyday lives providing motivation to develop an objective understanding of stress during typical activities. Physiological and physical response signals showing symptoms for stress can be used to provide hundreds of features. This encounters the problem of selecting appropriate features for stress recognition from a set of features that may include irrelevant, redundant or corrupted features. In addition, there is also a problem for selecting an appropriate computational classification model with optimal parameters to capture general stress patterns. The aim of this paper is to determine whether stress can be detected from individual-independent computational classification models with a genetic algorithm (GA) optimization scheme from sensor sourced stress response signals induced by reading text. The GA was used to select stress features, select a type of classifier and optimize the classifier's parameters for stress recognition. The classification models used were artificial neural networks (ANNs) and support vector machines (SVMs). Stress recognition rates obtained from an ANN and a SVM without a GA were 68% and 67% respectively. With a GA hybrid, the stress recognition rate improved to 89%. The improvement shows that a GA has the capacity to select salient stress features and define an optimal classification model with optimized parameter settings for stress recognition.

Keywords: stress classification, artificial neural networks, genetic algorithms, support vector machines, reading.

1 Introduction

Stress is part of everyday life and it has been widely accepted that stress, which can lead to less favorable emotional states, such as anxiety, fear or anger, is a growing concern for people and society. Stress has been defined as "the non-specific response of the body to any demand for change" [1]. It is the body's response to the imbalance caused between demands and resources available to a person. Stress is seen as a natural alarm, resistance and exhaustion [2] system for the body to prepare for a fight or flight response to protect the body from threats and changes. When experienced for longer periods of time without being managed, stress has been widely recognized as a major growing concern. It has the potential to cause chronic illnesses (e.g. cardiovascular

L. Vanneschi, W.S. Bush, and M. Giacobini (Eds.): EvoBIO 2013, LNCS 7833, pp. 117–128, 2013.

diseases, diabetes and some forms of cancer) and increase economic costs in societies, especially in developed countries [3, 4]. Benefits of stress research range from improving day-to-day activities, through increasing work productivity to benefitting the wider society - motivating interest, making it a beneficial area of research and posing technical challenges in Computer Science.

Objectively, stress has been interpreted from the human body's hormonal imbalances and response signals obtained from non-invasive methods. When a person is under stress, increased amounts of stress hormones (e.g. cortisol or catecholamine levels) are released and measures for these hormones are obtained via invasive methods (e.g. taking blood, saliva or urine samples) performed by qualified practitioners, and require lengthy analysis procedures conducted by qualified scientists [5-8]. On the other hand, the response signals that can be captured by non-invasive methods are easier to acquire, relatively cheaper and analysis time periods are relatively shorter. These are some of the reasons why non-invasive methods for objective stress detection are popular in literature [9-14]. Response signals are known to reflect reactions of individuals and their bodies to stressful situations [15, 16]. Stress response signals used in this paper fall into two categories – physiological and physical signals. Physiological signals include the galvanic skin response (GSR), electrocardiogram (ECG) and blood pressure (BP). Unlike these signals, we define physical signals as signals where changes by the human body can be seen by humans without the need for equipment and tools that need to be attached to individuals to detect general fluctuations. However, sophisticated equipment and sensors using vision technologies are still needed to obtain physical signals at sampling rates sufficient for data analysis and modeling like the ones used in this paper. Physical signals include eye gaze and pupil dilation signals. GSR, ECG, BP, eye gaze and pupil dilation signals have been used to detect stress in literature [13, 17, 18] but this combination has not been reported in literature so far. We use this combination of sensor signals in this paper and refer to them as primary signals for stress.

Various computational methods have been used to objectively classify stress to differentiate conditions causing stress from other conditions. The methods developed have used somewhat simplistic models based on techniques like Bayesian networks [18], decision trees [19] and support vector machines [20]. The parameters for the stress models in literature were chosen using a trial-and-error process. In this paper, models based on support vector machines (SVMs) and artificial neural networks (ANNs) are used and a genetic algorithm (GA) method is proposed for optimizing parameters for the models and selecting the better model for stress classification.

Further, computational models for stress recognition developed in literature have used a relatively smaller set of stress features than the sets used for the models in this paper [18-20]. Hundreds of stress features can be derived from primary signals for stress detection. However, this set of features may include redundant and irrelevant features which may outweigh the more effective features showing stress patterns. This could cause a stress classifier to produce lower quality classifications. Since this paper is dealing with sensor data, some features may suffer from corruption as well. In order to achieve a good classification model that is robust to such features that may reduce the performance of classifications, appropriate feature selection methods must be

developed and adopted by classifiers. A GA could be used to select subsets of features for optimizing stress classifications. It is a global search algorithm and has been commonly used to solve optimization problems [21] including feature selection problems to select features derived from physiological signals [22, 23]. Approaches used in this paper use a GA to select appropriate stress features with the goal to improve the quality for stress classification.

The performance for classification models not only depends on the inputs provided but also on model parameter settings. Hybrid GAs have been presented in literature that selected features for SVM based classification and optimized SVM parameters for various applications such as bankruptcy prediction [24], microarray analysis [25], intrusion detection [26]. GA and ANN hybrid methods with feature selection and parameter optimization has also been developed (e.g. weight optimization [27]) including applications in gear fault detection [28] and retail credit risk assessment [29]. The methods used in literature used different applications, different parameters for optimization and different features for selection from the ones presented in this work. Moreover, these forms of hybrid GAs are novel to the area of stress research.

This paper details the reading experiment done for stress data acquisition. It proposes hybrid GA methods to select an appropriate stress classification model, optimize parameter settings for the model to best capture stress patterns and select an appropriate subset of features as inputs for the model for stress recognition. Results for the performance of the hybrid GA methods are presented and discussed in regards to the aim for this work, which is to determine whether GA hybrid methods can develop stress classification models that improve the quality for recognizing stress patterns. The paper concludes with a summary of the findings and some suggestions for future work.

2 Stress Data Acquisition from Reading Experiment

Stress data was collected from a reading experiment where experiment participants read various types of text. Thirty-five undergraduate students were recruited as experiment participants. The participant cohort was made up of 25 males and 10 females over the age of 18 years old. Each participant had to understand the requirements of the experiment from a written set of experiment instructions with the guidance of the experiment instructor before they provided their consent to take part in the experiment. Afterwards, physiological stress sensors were attached to the participant and physical stress sensors were calibrated. The instructor notified the participant to start reading, which triggered a sequence of text paragraphs. After finishing the reading, participants had to do an assessment based on the reading. An outline of the process of the experiment for an experiment participant is shown in Fig. 1.

Each participant had physiological and physical measurements taken over the 12 minutes reading time period. During the reading period, a participant read *stressed* and *non-stressed* types of text. Stressed text had stressful content in the direction towards distress, fear and tension whereas the non-stressed text had content that created an illusion of meditation or soothing environments validated by participants. Each

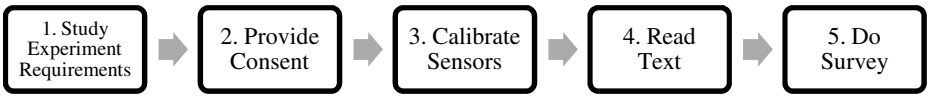


Fig. 1. Process followed by an experiment participant during the reading experiment

type of text had the same number of paragraphs and each paragraph was displayed on a computer monitor for participants to read one at a time. A participant read three stressed and three non-stressed paragraphs in some order. For consistency, each paragraph was displayed on a 1050 x 1680 pixel Dell monitor, displayed for 60 seconds and positioned at the same location of the computer screen for each participant. Each line of the paragraph had 70 characters including spaces.

Results from the experiment survey validated the text classes. This is a common method used in literature to validate stress classes for tasks [30]. Participants found the paragraphs that were labeled stressed as stressful and text labeled non-stressed as not stressful with a statistical significance of $p < 0.001$ according to the Wilcoxon test.

Physiological and physical sensor signals (which we refer to as primary stress signals) captured during the experiment were GSR, ECG, BP, eye gaze and pupil diameter signals. Biopac ECG100C, Biopac GSR100C and Finapres Finger Cuff systems were used to take ECG, GSR and blood pressure recordings at a sampling rate of 1000 Hz. Eye gaze and pupil dilation signals were obtained using Seeing Machines FaceLAB system with a pair of infrared cameras at 60 Hz. A schematic diagram of the experiment setup is shown in Fig. 2.

There were other stress signals that were derived from the primary stress signals to form other stress response signals. These signals included the heart rate variability signal, which was calculated from consecutive ECG peaks. The heart rate variability signal is another popular signal used for stress detection [9, 31].

Features were derived from stress signals. Statistics (including mean and standard deviation) were calculated for the signal measurements for each 5 second interval during the stressed and non-stressed reading. Measures such as the number of peaks for periodic signals, the distance an eye covered, the number of forward and backward tracking fixations, and the proportion of the time the eye fixated on different regions of the computer screen over 5 second intervals were also obtained. The statistic and measure values formed the stress feature set. There were 215 features in total.

3 Hybrid Genetic Algorithm, Artificial Neural Network and Support Vector Machine Stress Classification Models

GAs have been widely used as a search algorithm for a wide range of optimization problems [23]. A GA is a global search algorithm and is inspired by the concept of natural evolution. It evolves a population of candidate solutions, represented by *chromosomes*, in search for better quality chromosomes. The search evolves a

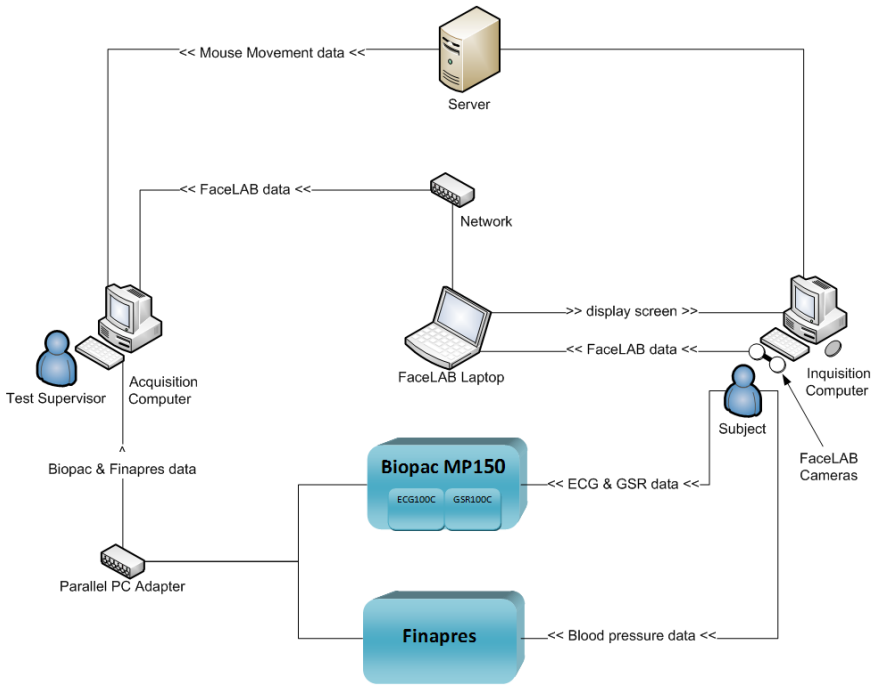


Fig. 2. Equipment setup for the reading experiment

population of chromosomes using *crossover*, *mutation* and *selection* methods. Crossover and mutation operations are applied to chromosomes to achieve diversity in the population and reduce the risk of the search being stuck with a local optimal population. After each generation during the search, the GA selects chromosomes, mostly made up of better quality chromosomes, for the population in the next generation to direct the search to more favorable chromosomes. In this paper, the quality of a chromosome is based on the accuracy, sensitivity, specificity and the F-score values of the stress classifications derived from the classifier, its parameters and a subset of stress features encoded by the chromosome. The classifiers used for stress classification were ANNs and SVMs.

An ANN, inspired by biological neural networks, has the capability to learn patterns to recognize characteristics in input tuples by classes. It is made up of interconnected processing elements, known as *artificial neurons*, which are connected by weighted links that pass signals between neurons. Feed-forward ANNs trained with the Levenberg-Marquardt learning algorithm were used in this paper.

The layers and neurons in each layer define the *topology* of a feed-forward ANN. An ANN has an input layer, may have multiple hidden layers and an output layer. Input tuples for the ANN are passed through the input layer. Then the weighted links pass the weighted signals to the neurons in the hidden layers which process the signals further and then in a similar fashion the signals progress through the following layers. The processed signals are propagated through the ANN in this fashion for the

signals to reach neurons in the output layer which causes the ANN to produce the output for the ANN. Essentially, a signal is propagated through the ANN layer after layer. In addition, the topology of the ANN can be designed to include a *time-delay* to exploit the time-varying nature of the stress features. Choosing the appropriate number of points in time for stress classification is another task that forms part of the process for designing the topology. The topology is usually designed by a trial and error process. Further, the topology may have an impact on the performance for classification. The approach proposed in this paper aims to optimize the topology for stress classification.

A SVM is another type of classification model that is used for stress classification in this paper. SVMs have been widely used in literature for classification problems including classifications based on physiological data [32]. Unlike an ANN, it is known to produce a global solution. Provided a set of training samples, a SVM transforms the data samples using a nonlinear mapping to a higher dimension with the aim to determine a *hyperplane* that partitions data by class or labels. The optimal hyperplane is chosen based on *support vectors*, which are training data samples that define maximum *margins* from the support vectors to the hyperplane to form the best decision boundary. This contributes to the resistance to data overfitting and helps to generalize classifications well. Further, there is a range of parameters that need to be chosen to define a SVM but this paper focuses on selecting and tuning the *kernel* function and the method for selecting the optimal hyperplane.

With the hundreds of stress features and the numerous options for settings for the stress classification models, a GA was used to find an optimal subset of features and search for a classifier that could best capture stress patterns in the features. Three different GAs, GA-ANN, GA-SVM and GA-ANN-SVM, were developed for stress classification. They differed in the type of classification model used to define stress recognition in the search and the type of chromosome. The chromosomes had components for features and parameters for classification models. Each type of chromosome was a binary string and represented the features and the parameters for the classification model. In the component of the chromosome that encoded the features, the index for a bit represented a feature and the bit value indicated whether the feature was used in the classification. For the components that encoded the parameter value, the binary string was treated as a binary number which was converted to its equivalent decimal number to extract the value for the component. The hybrid GAs are defined as follows:

GA-ANN. The GA used an ANN for stress classification. The chromosome encoded the following components:

- Stress features
- Number of hidden layers (ANN)
- Number of neurons in the first hidden layer (ANN)
- Number of neurons in the second hidden layer (ANN)
- Number of neurons in the third hidden layer (ANN)
- Time-delay (ANN)

GA-SVM. The GA used a SVM for stress classification. The chromosome encoded the following components:

- Stress features
- Type of kernel function: The bit value represented polynomial or Gaussian radial basis function. (SVM)
- Type of method: The bit value represented quadratic programming or sequential minimal optimization method. (SVM)
- Degree for the polynomial kernel function (SVM)
- Sigma value for the Gaussian radial basis function (SVM)

GA-ANN-SVM. The GA used an ANN or a SVM for stress classification depending on the value for chromosome component that encoded the type of classification model. The chromosome in this search is essentially made up of the chromosomes in GA-ANN and GA-SVM and encoded the following components:

- Stress features
- Type of classification model: The bit value represented ANN or SVM.
- Number of hidden layers (ANN)
- Number of neurons in the first hidden layer (ANN)
- Number of neurons in the second hidden layer (ANN)
- Number of neurons in the third hidden layer (ANN)
- Time-delay (ANN)
- Type of kernel function: The bit value represented polynomial or Gaussian radial basis function. (SVM)
- Type of method: The bit value represented quadratic programming or sequential minimal optimization method. (SVM)
- Degree for the polynomial kernel function (SVM)
- Sigma value for the Gaussian radial basis function (SVM)

Some components in the chromosomes for the GA hybrids are analogous to recessive genes in the biological domain. For GA-ANN, the chromosome component that provided the value for the number of hidden layers was used to determine whether the chromosome components that encoded the number of hidden neurons for each of the hidden layers were needed to be interpreted. For instance, if the number of hidden layers was interpreted to be 2, then the components that encoded the number of neurons in the first and second hidden layers were interpreted to define the ANN for classification and the component that encoded the number of neurons for the third hidden layer was not used. Likewise for GA-SVM, the component for the degree for the polynomial function was interpreted only if the polynomial function was selected as the kernel function i.e. the value for component that encoded the type of the kernel function was the polynomial function. Alternatively if the Gaussian radial basis function was selected, the sigma value for the Gaussian radial basis function was interpreted to define the SVM. GA-ANN-SVM was implemented in a similar fashion with the addition of the chromosome component that provided which type of classifier to use. If ANN was selected, then only the components that encoded the settings for ANN were

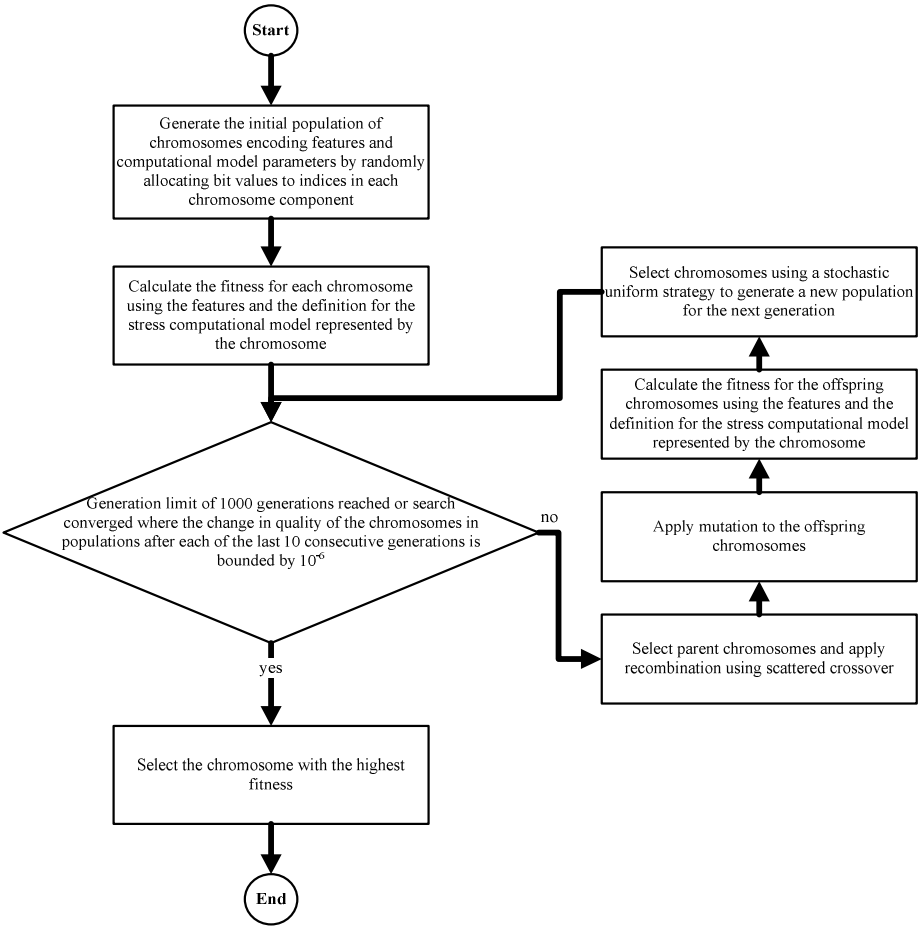


Fig. 3. The architecture for a GA hybrid algorithm for stress classification

Table 1. Parameter settings for the GA

GA Parameter	Value/Setting
population size	100
number of generations	2000
crossover rate	0.8
mutation rate	0.01
crossover type	MATLAB's Scattered Crossover
mutation type	MATLAB's Uniform Mutation
selection type	MATLAB's Stochastic Uniform Selection

interpreted and the components for SVM settings were not interpreted and similarly if SVM was selected then the SVM setting components were interpreted and ANN setting components were not interpreted.

The architecture for the GA hybrid algorithms is shown in Fig. 3 and Table 1 gives the parameter settings for the GA used in the search implementation.

4 Results and Discussion

Hybrid GAs were developed and tested on the reading data set for stress classification. The stress classification results achieved by an ANN, SVM and the different GA hybrids based on ANNs and SVMs are presented in Table 2. Ten-fold cross-validation was used to calculate the performance for the classifications. Overall, the GA hybrids produced better classification results than methods that did not use a GA to select appropriate stress features and optimize parameter settings for classification models to capture stress patterns. GA-ANN, GA-SVM and GA-ANN-SVM produced the best classification results with recognition rates of 89% based on two significant figures. It suggests that this recognition rate is the best result achievable for stress classification in reading based on an ANN or SVM. The recognition rates for GA hybrids were at least 21% greater than the rates produced by classification methods without a GA.

Table 2. Performance for the stress classification methods using ten-fold cross-validation

Classification Performance Measure	ANN	SVM	GA-ANN	GA-SVM	GA-ANN-SVM
Recognition rate	0.68	0.67	**0.89**	**0.89**	**0.89**
F-score	0.67	0.67	0.87	**0.89**	0.88

Table 3. Execution times for the stress classification methods

	ANN	SVM	GA-ANN	GA-SVM	GA-ANN-SVM
Execution time (hours)	0.05	0.01	96.03	5.12	45.82

Performance measures for the classification methods show that it was beneficial to use the feature selection and classifier parameter optimization methods in order to develop a classification model that captured stress patterns in reading. Using all the features for stress classification, like for the ANN and the SVM methods, would have included redundant and irrelevant features which may have outweighed the important features for stress recognition. The GA hybrids reduced the redundant and irrelevant features to develop stress classifiers.

Execution times were recorded for the methods while they were tested on the reading data set. The execution times for each method are shown in Table 3. GA-ANN

and GA-ANN-SVM took several days to produce a result, which was a lot longer than the other methods. The execution time for GA-SVM was in the order of hours.

In terms of quality of stress classifications produced and the amount of time taken to produce a result, GA-SVM performed the best out of all the other classification methods with the highest stress recognition rate and relatively low search execution time.

5 Conclusion and Future Work

GA hybrids that selected appropriate stress features and optimized classification model parameters were developed and tested for stress recognition. There were three different GAs proposed and they differed in the classification models which were used to define quality of stress classification in the search. One of the GAs used an ANN and incorporated an optimization strategy for the ANN to classify stress using features selected by the search (GA-ANN). Another GA hybrid was defined similarly but an SVM classifier was used instead of the ANN classifier (GA-SVM). The other GA hybrid was defined to select the type of classifier out of ANN and SVM to determine the quality of stress classification as well as select appropriate stress features and optimize classifier parameter settings (GA-ANN-SVM). The GA hybrids were tested on the reading data set to recognize stress patterns. Results showed that the quality for stress classification based on the ANN and SVM classifications improved with their GA hybrid counterparts – GA-SVM and GA-ANN produced better classification results than SVM and ANN respectively. In future, this work could be extended to incorporate a mechanism to select salient time segments in reading to determine critical time segments that are needed to differentiate the different reading classes in terms of stress measures. This approach could improve the performance for recognizing stress in reading.

References

[1] Selye, H.: The stress syndrome. The American Journal of Nursing 65, 97–99 (1965)
[2] Hoffman-Goetz, L., Pedersen, B.K.: Exercise and the immune system: a model of the stress response? Immunology Today 15, 382–387 (1994)
[3] The-American-Institute-of-Stress. America's No. 1 Health Problem - Why is there more stress today? (August 05, 2010), http://www.stress.org/americas.htm
[4] Lifeline-Australia, Stress Costs Taxpayer $300K Every Day (2009), http://www.lifeline.org.au
[5] Jin, P.: Efficacy of tai chi, brisk walking, meditation, and reading in reducing mental and emotional stress. Journal of Psychosomatic Research 36, 361–370 (1992)
[6] Lerner, J.S., Dahl, R.E., Hariri, A.R., Taylor, S.E.: Facial expressions of emotion reveal neuroendocrine and cardiovascular stress responses. Biological Psychiatry 61, 253–260 (2007)
[7] Ritter, F.E., Ceballos, R., Reifers, A.L., Klein, L.C.: Measuring the effect of dental work as a stressor on cognition (Tech. Report No. 2005-1): Applied Cognitive Science Lab, School of Information Sciences and Technology, Penn State (2005), http://acs.ist.psu.edu/acslab/reports/ritterCRK05.pdf

[8] Lundberg, U., Kadefors, R., Melin, B., Palmerud, G., Hassmén, P., Engström, M., Elfsberg Dohns, I.: Psychophysiological stress and EMG activity of the trapezius muscle. International Journal of Behavioral Medicine 1, 354–370 (1994)
[9] Dishman, R.K., Nakamura, Y., Garcia, M.E., Thompson, R.W., Dunn, A.L., Blair, S.N.: Heart rate variability, trait anxiety, and perceived stress among physically fit men and women. International Journal of Psychophysiology 37, 121–133 (2000)
[10] Steptoe, A., Marmot, M.: Impaired cardiovascular recovery following stress predicts 3-year increases in blood pressure. Journal of Hypertension 23, 529 (2005)
[11] Partala, T., Surakka, V.: Pupil size variation as an indication of affective processing. International Journal of Human-Computer Studies 59, 185–198 (2003)
[12] Lundberg, U.: Stress, subjective and objective health. International Journal of Social Welfare 15, S41–S48 (2006)
[13] Labbé, E., Schmidt, N., Babin, J., Pharr, M.: Coping with stress: the effectiveness of different types of music. Applied Psychophysiology and Biofeedback 32, 163–168 (2007)
[14] Sharma, N., Gedeon, T.: Objective measures, sensors and computational techniques for stress recognition and classification: A survey. Computer Methods and Programs in Biomedicine 108, 1287–1301 (2012)
[15] Bakker, J., Pechenizkiy, M., Sidorova, N.: What's your current stress level? detection of stress patterns from gsr sensor data. In: International Conference on Data Mining Workshops (ICDMW), Vancouver, BC, pp. 573–580 (2011)
[16] Yuen, P., Hong, K., Chen, T., Tsitiridis, A., Kam, F., Jackman, J., James, D., Richardson, M., Williams, L., Oxford, W.: Emotional & Physical Stress Detection and Classification Using Thermal Imaging Technique. In: 3rd International Conference on Crime Detection and Prevention (ICDP 2009), London, pp. 1–6 (2009)
[17] Healey, J.A., Picard, R.W.: Detecting stress during real-world driving tasks using physiological sensors. IEEE Transactions on Intelligent Transportation Systems 6, 156–166 (2005)
[18] Liao, W., Zhang, W., Zhu, Z., Ji, Q.: A real-time human stress monitoring system using dynamic bayesian network. In: Computer Vision and Pattern Recognition - Workshops, CVPR Workshops (2005)
[19] Zhai, J., Barreto, A.: Stress recognition using non-invasive technology. In: Proceedings of the 19th International Florida Artificial Intelligence Research Society Conference, FLAIRS, pp. 395–400 (2006)
[20] Dou, Q.: An SVM ranking approach to stress assignment. University of Alberta (2009)
[21] Goldberg, D.E.: Genetic algorithms in search, optimization and machine learning. Addison-wesley (1989)
[22] Park, B.J., Jang, E.H., Kim, S.H., Huh, C., Sohn, J.H.: Feature selection on multiphysiological signals for emotion recognition. In: 2011 International Conference on Engineering and Industries (ICEI), Korea, pp. 1–6 (2011)
[23] Niu, X., Chen, L., Chen, Q.: Research on genetic algorithm based on emotion recognition using physiological signals. In: International Conference on Computational Problem-Solving, pp. 614–618 (2011)
[24] Min, S.H., Lee, J., Han, I.: Hybrid genetic algorithms and support vector machines for bankruptcy prediction. Expert Systems with Applications 31, 652–660 (2006)
[25] Mohamad, M.S., Deris, S., Illias, R.M.: A hybrid of genetic algorithm and support vector machine for features selection and classification of gene expression microarray. International Journal of Computational Intelligence and Applications 5, 91–107 (2005)

[26] Kim, D.S., Nguyen, H.N., Park, J.S.: Genetic algorithm to improve SVM based network intrusion detection system. In: International Conference on Advanced Information Networking and Applications, pp. 155–158 (2005)
[27] Jadav, K., Panchal, M.: Optimizing Weights of Artificial Neural Networks using Genetic Algorithms. International Journal of Advanced Research in Computer Science and Electronics Engineering (IJARCSEE) 1, 47–51 (2012)
[28] Samanta, B., Al-Balushi, K., Al-Araimi, S.: Artificial neural networks and support vector machines with genetic algorithm for bearing fault detection. Engineering Applications of Artificial Intelligence 16, 657–665 (2003)
[29] Oreski, S., Oreski, D., Oreski, G.: Hybrid system with genetic algorithm and artificial neural networks and its application to retail credit risk assessment. Expert Systems with Applications 39, 12605–12617 (2012)
[30] Hill, J.D., Boyle, L.N.: Driver stress as influenced by driving maneuvers and roadway conditions. Transportation Research Part F: Traffic Psychology and Behaviour 10, 177–186 (2007)
[31] Ferreira, P., Sanches, P., Höök, K., Jaensson, T.: License to chill!: how to empower users to cope with stress. In: Proceedings of the 5th Nordic Conference on Human-Computer Interaction: Building Bridges, pp. 123–132 (2008)
[32] Cheng, B.: Emotion Recognition from Physiological Signals Using Support Vector Machine. In: Wu, Y. (ed.) Software Engineering and Knowledge Engineering. AISC, vol. 114, pp. 49–52. Springer, Heidelberg (2012)

Optimal Use of Biological Expert Knowledge from Literature Mining in Ant Colony Optimization for Analysis of Epistasis in Human Disease

Arvis Sulovari, Jeff Kiralis, and Jason H. Moore

Dartmouth-Hitchcock Medical Center, Lebanon
New Hampshire, 03756, United States
arvissulovari@gmail.com, {Jason.H.Moore,Jeff.Kiralis}@dartmouth.edu
http://www.epistasis.org

Abstract. The fast measurement of millions of sequence variations across the genome is possible with the current technology. As a result, a difficult challenge arise in bioinformatics: the identification of combinations of interacting DNA sequence variations predictive of common disease [1]. The Multifactor Dimensionality Reduction (MDR) method is capable of analysing such interactions but an exhaustive MDR search would require exponential time. Thus, we use the Ant Colony Optimization (ACO) as a stochastic wrapper. It has been shown by Greene et al. that this approach, if expert knowledge is incorporated, is effective for analysing large amounts of genetic variation[2]. In the ACO method integrated in the MDR package, a linear and an exponential probability distribution function can be used to weigh the expert knowledge. We generate our biological expert knowledge from a network of gene-gene interactions produced by a literature mining platform, Pathway Studio. We show that the linear distribution function of expert knowledge is the most appropriate to weigh our scores when expert knowledge from literature mining is used. We find that ACO parameters significantly affect the power of the method and we suggest values for these parameters that can be used to optimize MDR in Genome Wide Association Studies that use biological expert knowledge.

1 Introduction

Human geneticists are now able to measure millions of DNA sequence variations across large patient sample datasets. These large datasets present a challenge in the field of informatics: which variations can be used to predict susceptibility to common human disease such as cancer? What makes this challenge even more difficult is the fact that susceptibility to a given disease cannot always be determined by the action of a single gene, but rather the action of multiple interacting genes. Moore argues that non additive interactions, known as epistasis, are likely to be ubiquitous in common human disease [2]. Moore's argument relies on four

L. Vanneschi, W.S. Bush, and M. Giacobini (Eds.): EvoBIO 2013, LNCS 7833, pp. 129–140, 2013.

important concepts: the notion of epistasis is grounded in almost one century of scientific literature, molecular interactions between proteins are ubiquitous in biological systems, a single locus model is insufficient for explaining the etiology of common human diseases, and when scientists have tried to find epistasis using powerful computational and biostatistical methods, they have often been able to find examples of it. If we want to find predictors of common human disease, we need to employ methods which take into consideration the complexity of biological systems.

Data from biological systems is noisy due to the inherent complexity of these systems. The noise is primarily due to the fact that disease states of subjects with the same values for the relevant attributes could be different. Moreover, the fitness landscape is rugged because the models that contain less than all of the relevant attributes may perform worse than the surrounding noise [1].

The International HapMap Consortium suggests that approximately $3 \cdot 10^6$ carefully selected SNPs (i.e. single nucleotide polymorphisms) may be necessary and sufficient to capture all variation among the human population [3]. If this were true, we would expect $\binom{3 \cdot 10^6}{2} = 4.5 \cdot 10^{12}$ potential epistatic pairwise interactions. Biological systems provide inspiration for much more efficient machine learning algorithms.

Greene et al. have shown that the ACO method can be used effectively for human genetics problems when expert knowledge is used [1]. We used biological expert knowledge extracted from literature mining and rigorously examined the two different weighing functions of expert knowledge within the ant colony system in MDR to suggest good parameters for later use in Genome Wide Association Studies (GWAS). We believe that the approach of using knowledge from literature mining to facilitate MDR's quest in finding epistatic models underlying common human disease has not been explored before. Most importantly, this method has potential to provide more biologically relevant findings with regard to epistasis than previous MDR approaches.

2 Literature Mining Using Pathway Studio

Pathway Studio is a software application developed for navigation and analysis of biological pathways by Ariadne Genomics [4]. This software comes with a database of more than 100,000 interaction types, regulation and modification events between proteins, cell processes and small molecules. The database has been compiled by MedScan, a text-mining tool, to the whole PubMed. MedScan pre-processes text input from the user to extract the relevant sentences which are then subjected to Natural Language Processing (NLP). The pre-processing step uses a manually curated biological dictionary of synonyms. The NLP kernel deduces the syntactic structure of the sentences and establishes logical relationship between concepts. Finally, the results are matched against the functional ontology to produce biologically interpretable data [4].

Here we queried all the genes corresponding to the SNPs in our dataset (Section 5). The output from Pathway Studio provided us with information on the

number of interactions for each gene. The number of connections for each gene was averaged across all the present types of interactions to give an expert knowledge score. This method represents one way of processing the biological knowledge from Pathway Studio into expert knowledge recognized by the ACO method in MDR. Our processing method considered SNPs which belonged to genes with many interactions as more important than those with less interactions. The ant system integrated into MDR used Pathway Studio as its source of expert knowledge.

3 Multifactor Dimensionality Reduction Platform

Greene et al. developed an ACO framework to be available in version 2.0 and later of the Multifactor Dimensionality Reduction (MDR) software package [1]. This package provides a user friendly cross-platform Java GUI appropriate for genome-wide genetic analysis. In short, MDR groups multilocus genotypes in high-risk and low-risk groups, reducing the genotype predictors' dimensionality from n to *1*. The new one dimensional multilocus-genotype variable is evaluated for its ability to classify and predict case-control status through cross-validation. The MDR method has been developed as a non-parametric and model-free genetic data mining strategy for identifying combinations of SNPs that are predictive of discrete clinical endpoint [7]. The MDR method has been successfully applied to detect gene-gene interactions in a variety of human diseases: breast cancer [7], type 2 diabetes [9], rheumatoid arthritis [8], and coronary artery disease [10].The MDR method is described in detail by Moore, et al. [5].

4 The MDR Ant Colony Optimization (ACO) Approach

The idea of using ants as an inspiration for machine learning algorithms is not new. Dorigo showed in 1991 that ants could be used as a search strategy by providing positive feedback [17]. In an ant system ants explore the landscape of possible solutions by leaving a trace of pheromones on each solution they find, depending on the quality of that discovery. Over time the pheromones evaporate and their signal weakens. The quantity of pheromone left on each discovery made by an ant determines the likelihood that the same region will be explored in the future by other ants. Dorigo and Stützle discuss how incorporation of a priori information can be used to derive heuristic information that biases the probabilistic decision taken by the ants [18]. ACO is one of the techniques of swarm intelligence, a relatively new domain within AI research, that has proven to be competitive with traditional techniques of data mining [19]. Moore et al. discovered that incorporation of a priori knowledge into machine learning algorithms is crucial if these algorithms are to succeed at genome-wide genetic analysis [20].

In the ant system integrated within the MDR package, the goal is to select the SNPs (i.e. attributes) which effectively determine an individual's risk of disease. We use Pathway Studio scores (Section 2) as biological expert knowledge.

The ACO method allows the user to select an exponential or linear function for weighing the scores. Below we discuss each of the resulting probability distributions. We assume that there are n attributes $A_1, \ldots, A_n$ with A_i having expert knowledge score S_i. We label the attributes so that $S_1 \leq S_2 \leq \ldots \leq S_n$.

4.1 Exponential Weighing

With exponential weighing, the probability that attribute A_i is selected is given by the exponential function [1]

$$P(A_i) = \frac{1}{\sum_{k=1}^{N} \theta^{-S_k}} \theta^{-S_i}, \tag{1}$$

where θ is the user-adjustable parameter, satisfying $0 < \theta \leq 1$, and here the expert knowledge scores S_i are normalized so that they lie between 0 and 2.

As Greene et al. noted, if θ is near 1, attributes with a high expert knowledge score are only slightly more likely to be chosen than those with a lower score. Otherwise, for instance, if θ is 1/3, the attributes with a high score are much more likely to be chosen than those with lower scores.

4.2 Linear Weighing

For linear weighing the probability that attribute S_i is selected is given by

$$P(A_i) = mS_i + b \tag{2}$$

for some constants m and b. We require that $m \geq 0$ so that $P(A_1) \leq P(A_2) \leq \ldots \leq P(A_n)$. This assures that attributes with larger expert knowledge scores are more apt to be selected.

The constraints $\sum_{i=1}^{n} P(A_i) = 1$ and $P(A_1) \geq 0$, and the requirement $m \geq 0$ are satisfied only when:

$$m \in \left[0, \frac{1}{\sum_{i=1}^{n}(S_i - S_1)}\right] \quad \text{and} \quad P(A_N) \in \left[\frac{1}{n}, \frac{S_N - S_1}{\sum_{i=1}^{n}(S_i - S_1)}\right].$$

Here $P(A_n)$ is the probability of selecting the attribute with the highest expert knowledge score.

The parameter $M_p \in [0, 1]$ adjusts m and $P(A_n)$ so that:

$$m = \frac{M_p}{\sum_{i=1}^{N}(S_i - S_1)} \quad \text{and} \quad P(A_n) = \frac{1}{N} + M_p \left(\frac{S_i - S_1}{\sum_{i=1}^{N}(S_i - S_1)} - \frac{1}{N}\right).$$

Both of these functions have the following pheromone update procedure:

$$\delta\tau_{a,i} = \sum_{k=1}^{m} Q_{a,b} \cdot S_a^{\beta}$$

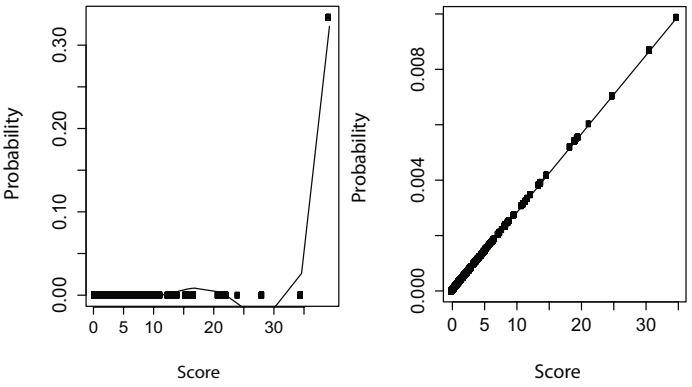


Fig. 1. Assignment of probabilities for each expert knowledge score being chosen according to the exponential distribution (left panel) and linear distribution (right panel). In the left panel, the probability of being chosen of any score below the maximum score is zero, which makes this function inappropriate for our expert knowledge. On the other hand, the linear probability distribution function assigns non-zero probabilities to many more attributes. The solid line on the left represents a polynomial best fit.

where $\delta\tau_{a,i}$ is the change in pheromone strength between updates. $Q_{a,b}$ is the MDR accuracy for a model containing both attributes a and b, while S_a is the biological expert knowledge from Pathway Studio for attribute a and β is a weighing exponent for the expert knowledge, E_a.

Figure 1 shows the distribution of probabilities for each expert knowledge score according to both distribution functions.

5 Data Simulation and Analysis

A genotype study conducted by Andrews et al. produced a SNP dataset of 1421 SNPs in approximately 400 hypothesized cancer-related genes from the SNP500 database [11]. This dataset contains 893 controls and 617 subjects with bladder cancer. Here we replaced a random set of 100 SNP-SNP pairs from the original dataset along with the class values (i.e. case/control status) with two synthetic epistatic SNPs and their respective new class values.

The generation of the synthetic SNPs and their class values was done using the GAMETES algorithm [12] [13]. This algorithm generates SNP datasets of various population sizes, heritabilites and allele frequencies. For every run of the GAMETES, there were three different models, in each one of which we had one epistatic SNP pair.

In our case, we generated nine models of disease risk, each containing two relevant epistatic SNPs. These models spanned three heritabilities (0.05, 0.1, and 0.2). For each heritability GAMETES generated three models. All these models exhibit no main effects when the SNPs have a minor allele frequency of 0.4, which were the conditions we used for generating our data. This means that

the effects in each dataset will be due to epistatic interactions and not main effects caused by a single SNP. We generated 100 datasets for each model.

In the modified datasets containing the synthetic SNPs, the noise was provided by the 1419 biological SNPs. Here we mapped Pathway Studio's expert knowledge scores for each gene, from a total of 397 genes, into expert knowledge scores for each respective SNP. As for the synthetic SNPs, we assigned them three different scores: upper 10%, upper 1%, and upper 0.1% cut-off values according to the overall ranking of the scores. We then provided these scores to the ant system which converted them into selection probabilities using the linear distribution function. (See Section 4.2)

We explored five major parameters of the ant system: maximum probability, β, retention factor, and number of ants and updates. Maximum probability (i.e. the slope of the linear probability function) was assigned values of 10%, 50% and 90%. Beta was assigned four different values: 0, 1, 2, and 4. Ants and updates were each assigned values of 100, 200, 400, and 800. The retention factor determines how much weight is given to information from the previous iterations relative to the most recent iteration. We considered retention factors of 0.1, 0.5 and 0.9. A total of 640,000 parameter combinations was explored. We considered a high number of total interactions between parameters in order to assure the discovery of the two epistatic SNPs. The sweep of all MDR parameters was done on a 1300-processor cluster at Dartmouth College. To determine statistical significance, we used logistic regression. Logistic regression allows for a rigorous examination of the effect that one or more continuous factors (i.e. parameters) have on the success rate, the number of runs that selected the two synthetic SNPs over the total runs. We used the R statistical programming language to run logistic regression [14]. We assessed all single and pairwise effects of

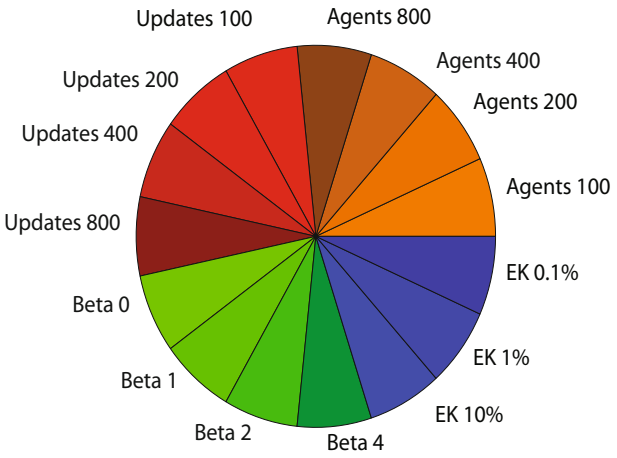


Fig. 2. Legend for the segment plots in Figure 3

all parameters on the success rate. Results of logistic regression were considered significant when $p \leq 0.05$.

6 Experimental Results

We found that a retention factor of 0.9 and a maximum probability of 90% are the best parameters for fine tuning the ACO method. These values support the findings by Greene et al. [1]. Hence, in the data presented below we fixed these two parameters and looked at combinations of the other ACO parameters: β, number of ants, number of updates, and the expert knowledge scores for the two

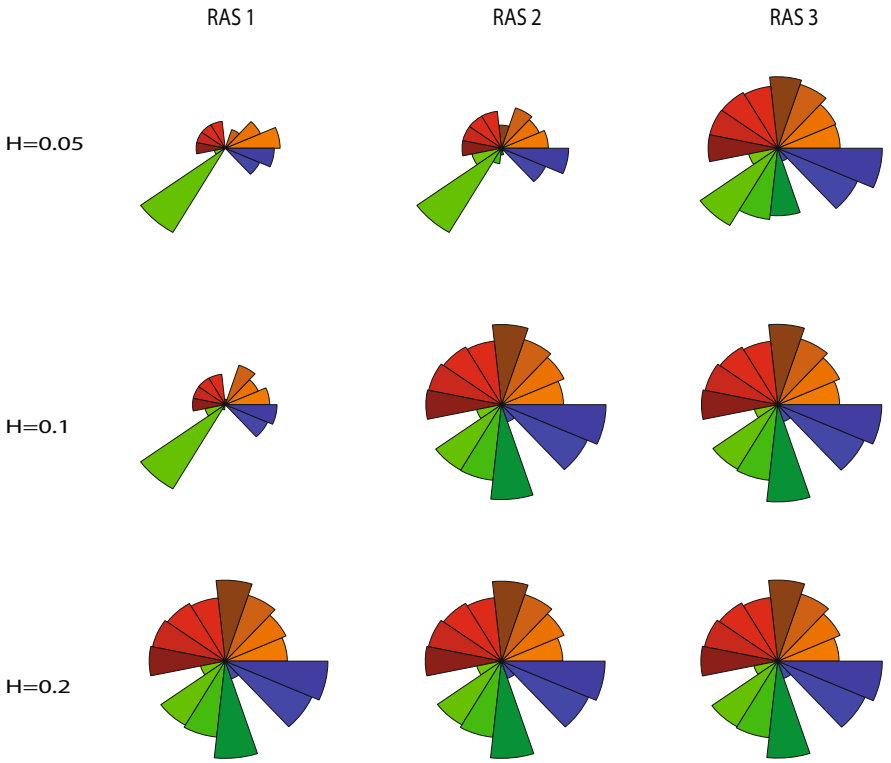


Fig. 3. Results of the simulations on all 9 models. Each plot summarizes the results from 19,200 sweep runs and the size of each sector in a plot represents the success rate of each respective parameter across all those runs. $Beta = 1$ is the most important parameter in the first four datasets along with the highest expert knowledge score. In the other 5 models, $Beta = 4$ yields the highest success along with the highest expert knowledge score and the highest number of ants. Each model was simulated by GAMETES under a Minor Allele Frequency of 0.4. Each row corresponds to a different heritability (H) and each column corresponds to a different quantile of the Relative Allele Signals (RAS).

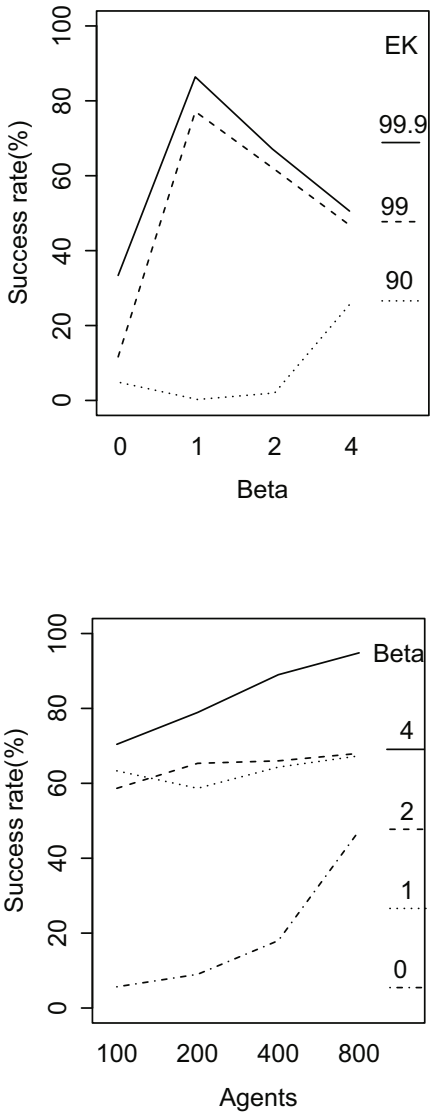


Fig. 4. Two of the pairwise parameter interaction plots that were found to be significant by logistic regression in models of heritability 0.05 (top panel) and 0.2 (bottom panel). The intersections or the non-parallel segments infer a pairwise interaction between the parameters which is also confirmed by logistic regression at a significance level of $p \leq 0.05$.

synthetic SNPs. There were a total of 19,200 runs for each model amounting to a 172,800 runs overall. The results of these runs per model are shown in Figures 2 and 3 [15]. Segment plots are used to visually summarize the results for each model. Although the exact contribution of each parameter to the overall success rate cannot be assessed, the relative contribution of each parameter within and between models can be easily determined. The highest expert knowledge score for the two epistatic SNPs (i.e. upper 0.1%) yielded the highest success rates across all datasets, which was to be expected. The number of agents had a slightly different behaviour. In the first two models of heritability of 0.05, the highest number of agents yielded the lowest success rate which seemed counter-intuitive at first. However, this behaviour was also to be expected since the signal of the synthetic SNPs was weaker than the biological noise in the first two models. Indeed, in the other seven models where the synthetic SNPs had a stronger signal, the highest number of ants yielded the highest success rate among the ant population sizes within models. Each of the four different ant update values had a near-uniform effect on the success rate. As for the expert knowledge weighing factor, β, we noticed an interesting behaviour. In all datasets with heritability of 0.05 and in the first two models with heritability of 0.1 for the synthetic SNPs, $\beta = 1$ had the highest effect on the success rate among the betas. This value of β was the same as the one suggested by Greene et al. [1]. However, in the remaining 5 models, $\beta = 4$ had the highest effect on the success rate among the betas. To understand better this behaviour of the β parameter, we ran MDR exhaustively on the first four models (data not shown). We found that our two synthetic SNPs were not the best two-way model chosen by MDR in the first four models, which were the same models where $\beta = 1$ had the highest success rate among the betas. The more exponential weight we put on our expert knowledge by increasing β, the lower the success rate became in each of the first four models where the synthetic SNPs seemed to be undetectable even by exhaustive MDR. This could be due to amplification of noise in those models when $\beta > 1$.

While segment plots can be very useful in visualizing vast amounts of data, they do not give us statistical details on the effect of single-parameter or pairwise interactions which is why we used logistic regression. Logistic regression showed that all single-parameter effects are significant at $p \leq 0.001$ across all three heritabilities with the exception of the number of ants and updates. These results agreed with the summary from the segment plots. Moreover, logistic regression showed that pairwise parameter interactions significant at $p \leq 0.05$ were β:expert knowledge in all models with heritability of 0.05, ants:expert knowledge in all models with heritability 0.1 and all possible pairwise parameter combinations in all models of heritability of 0.2 with the exception of ants:updates. To visualize the pairwise parameter interactions, we used interaction plots (Figure 4). Based on these results, we suggest parameter settings of $\beta = 1$, in the case of a weak epistatic signal and $\beta = 4$ otherwise, retention factor of 0.9 and maximum probability of 90%. The number of ants and updates would be best if set at the highest value given the computational constraints. Our recommended settings

for the β parameter are different from those of Greene et al. which is most likely due to the different source of expert knowledge and the fact that the noisy SNPs in our data are biological and not simulated.

7 Discussion

Our study highlights the importance of utilizing biological expert knowledge in guiding GWAS. Here we presented one method of integrating biological expert knowledge from Pathway Studio into the ACO algorithm within MDR. The interactions found by Pathway Studio in the literature have more biological relevance than those generated by statistical methods alone. Recent studies have emphasized the importance of using biologically relevant expert knowledge in computational methods attempting to detect epistasis in genome wide genetic analysis [6] [21]. Our approach can yield biologically relevant results as defined by the current literature.

We chose the linear function to weigh the expert knowledge scores extracted from Pathway Studio as it presented one important advantage over the exponential function: it assigned non-zero probabilities of being chosen to more attributes (i.e. SNPs) compared to the exponential function. The linear function guarantees us that MDR will explore a bigger space of the solutions' landscape and yet spend less time compared to an exhaustive run. The solutions considered in this landscape also have a high biological relevance due to the source of expert knowledge.

We observed several interactions between the ant system parameters. Both logistic regression and segment plots helped us understand and visualize the effect that each parameter as well as pairwise combinations of parameters had on the overall success rate of the ACO.

Alternatives to processing our expert knowledge from Pathway Studio have been considered. We could make the scores even more biologically relevant by calculating the expert knowledge scores for every pairwise interaction in our dataset instead of calculating them for every single SNP, in order to estimate the relevance of the interactions using mutual information scores. The latter approach would also require a modification of the current ACO method in MDR as it currently only accepts scores for individual SNPs. Another improvement can be done on the function used to weigh the expert knowledge scores. We chose the linear function because of its superior representation of scores over the exponential function. However, these two functions do not present the only two heuristics' probability functions that can be used. In fact, as Dorigo and Stützle discuss in their book, the ACO algorithm could have other additional features, such as the Model Based Search [18] which is yet to be explored.

Our understanding of common human disease would be enhanced if more methods which take into consideration biologically relevant knowledge, similar to the approach we have presented, can be developed to detect epistasis in GWAS. If the epistasis quest of computational methods, such as MDR, is facilitated and directed by biologically relevant knowledge, then our preventative, diagnostic

and treatment options will improve and could lead to better health and lower incidence of common disease.

Acknowledgements. The work was funded by grants R01 LM010098, LM009012 and AI59694 from the U.S.A National Institute of Health.

References

1. Greene, C.S., Gilmore, J.M., Kiralis, J., Andrews, P.C., Moore, J.H.: Optimal Use of Expert Knowledge in Ant Colony Optimization for the Analysis of Epistasis in Human Disease. In: Pizzuti, C., Ritchie, M.D., Giacobini, M. (eds.) EvoBIO 2009. LNCS, vol. 5483, pp. 92–103. Springer, Heidelberg (2009)
2. Moore, J.H.: The ubiquitous nature of epistasis in determining susceptibility to common human disease. Human Heredity 56, 73–82 (2003)
3. The International HapMap Consortium: A second Generation human haplotype of over 3.1 million SNPs. Nature 449, 851–861 (2007)
4. Nikitin, A., Egorov, S., Mazo, I.: Pathway Studio-the analysis and navigation of molecular networks. Bioinformatics Oxford Journals 19(16), 2155–2157 (2003)
5. Moore, J.H., Gilbert, J.C., Tsai, C.T., Chiang, F.T., Holden, T., Barney, N., White, B.C.: A flexible computational framework for detecting, characterizing, and interpreting statistical patterns of epistasis in genetic studies of human disease susceptibility. Journal of Theoretical Biology 241(2), 252–261 (2006)
6. Cordell, H.J.: Detecting gene-gene interactions that underlie human diseases. Nature Review Genetics 10, 392–404 (2009)
7. Ritchie, M.D., et al.: Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer. American Journal of Human Genetics 69, 138–147 (2001)
8. Julia, A., et al.: Identification of a two-loci epistatic interaction associated with susceptibility to rheumatoid arthritis through reverse engineering and multifactor dimensionality reduction. Genomics 90, 6–13 (2007)
9. Cho, Y.M., et al.: Multifactor-dimensionality reduction shows a two-locus interaction associated with type 2 diabetes mellitus. Diabetologia 47, 549–554 (2004)
10. Tsai, C.T., et al.: Reninangiotensin system gene polymorphisms and coronary artery disease in a large angiographic cohort: detection of high order genegene interaction. Atherosclerosis 195, 172–180 (2007)
11. Andrew, A.S., et al.: Bladder Cancer SNP panel predicts susceptibility and survival. Human Genetics 125(5-6), 527–539 (2009)
12. Urbanowicz, R.J., Kiralis, J., Sinnot-Armstrong, N.A., Heberling, T., Fisher, J.M., Moore, J.H.: GAMETES: a fast, direct algorithm for generating pure, strict, epistatic models with random architectures. BioData Mining 5(16) (2012)
13. `http://sourceforge.net/projects/gametes/`
14. `http://stat.ethz.ch/R-manual/R-patched/library/stats/html/glm.html`
15. `http://stat.ethz.ch/R-manual/R-patched/library/graphics/html/stars.html`
16. Sokal, R.R., Rohlf, F.J.: Biometry: the principles and practice of statistics in biological research, 3rd edn. W.H. Freeman and Co., New York (1995)
17. Dorigo, M., Maniezzo, V., Colorni, A.: Positive Feedback as a search strategy. Dipartimento di Elettronica e Informatica, Politecnico di Milano, Technical Reports, 91–116 (1991)

18. Dorigo, M., Stützle, T.: Ant Colony Optimization (2004)
19. Martens, D., et al.: Editorial Survey: Swarm Intelligence for Data Mining. Machine Learning 82(1), 1–42 (2011)
20. Moore, J.H., White, B.C.: Genome-wide genetic analysis using genetic programming: The critical need for expert knowledge. In: Genetic Programming Theory and Practice IV. Springer (2007)
21. Pattin, K., Moore, J.H.: Exploiting the proteome to improve the genome-wide genetic analysis of epistasis in common human diseases. Human Genetics 124(1), 19–29 (2008)

A Multiobjective Proposal Based on the Firefly Algorithm for Inferring Phylogenies

Sergio Santander-Jiménez and Miguel A. Vega-Rodríguez

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

Abstract. Recently, swarm intelligence algorithms have been applied successfully to a wide variety of optimization problems in Computational Biology. Phylogenetic inference represents one of the key research topics in this area. Throughout the years, controversy among biologists has arisen when dealing with this well-known problem, as different optimality criteria can give as a result discordant genealogical relationships. Current research efforts aim to apply multiobjective optimization techniques in order to infer phylogenies that represent a consensus between different principles. In this work, we apply a multiobjective swarm intelligence approach inspired by the behaviour of fireflies to tackle the phylogenetic inference problem according to two criteria: maximum parsimony and maximum likelihood. Experiments on four real nucleotide data sets show that this novel proposal can achieve promising results in comparison with other approaches from the state-of-the-art in Phylogenetics.

Keywords: Swarm Intelligence, Multiobjective Optimization, Phylogenetic Inference, Firefly Algorithm.

1 Introduction

Bioinformatics aims to address problems that imply the processing of a growing number of biological data by means of computational techniques. Most of these problems cannot be tackled by using exhaustive searches because of their NP-hard complexity. Additionally, these biological problems can be addressed from several, conflicting perspectives. Recent research works try to overcome such limitations by applying multiobjective metaheuristics [1]. Their main goal is to generate a set of Pareto solutions that represent a compromise between different criteria, by optimizing simultaneously two or more objective functions [2].

One of the key problems in Computational Biology is the reconstruction of ancestral genealogical relationships among species, phylogenetic inference [3]. Phylogenetic procedures take as input molecular characteristics of organisms, such as nucleotide sequences represented by using an alphabet $\Sigma = \{A, C, G, T\}$. By analyzing these sequences, we get a mathematical structure that describes a

L. Vanneschi, W.S. Bush, and M. Giacobini (Eds.): EvoBIO 2013, LNCS 7833, pp. 141–152, 2013.

hypothesis about the evolutionary history of these species, the phylogenetic tree. Input species represent the results of the evolutionary process and are located at the leaves of the tree. Hypothetical ancestors are represented by internal nodes, and ancestor-descendant relationships are modelled by branches.

In the literature we can find a variety of optimality criteria for inferring phylogenies, such as maximum parsimony, maximum likelihood and distance methods [3]. However, by using a specific criterion, the resulting phylogenies can be radically different to the trees generated by other criteria, inferring discordant genealogical relationships [4]. This fact motivates that biologists are forced to use several single-criterion software to analyze complex data sets, and publish results that make clear these conflicts. By means of multiobjective optimization, we try to support a complementary view of phylogenetic inference according to multiple criteria, with the aim of generating a set of phylogenetic trees that represent a consensus between different points of view.

In this paper we propose a multiobjective adaptation of the novel Firefly Algorithm (FA) for inferring phylogenetic trees attending to the parsimony and likelihood principles. We have chosen this swarm intelligence algorithm due to the promising results reported for a variety of problems, overcoming other bioinspired proposals [5]. In order to assess the performance of this Multiobjective Firefly Algorithm (MO-FA), we have carried out experiments on four nucleotide data sets, applying the hypervolume metrics [2], and comparing results with other authors' multiobjective proposals and popular single-criterion methods.

This paper is organized as follows. In the next section, we introduce a short review on other bioinspired proposals for inferring phylogenetic trees. In Section 3, we detail the basis of phylogenetic methods based on distances, parsimony and likelihood. Section 4 explains the details about MO-FA and discusses how to adapt it to multiobjective phylogenetic inference. Experimental results are presented and explained in Section 5, introducing comparisons with other proposals. Finally, Section 6 summarizes some conclusions and future research lines.

2 Related Work

Throughout the years, several bioinspired proposals have been published with the aim of carrying out phylogenetic analyses on data sets with a growing complexity. In such data sets, exhaustive searches cannot be applied due to the huge number of possible phylogenetic topologies, which increases in an exponential way with the number of species [3]. In this section, we summarize several bioinspired approaches to Phylogenetics proposed by other authors.

The first bioinspired proposals for inferring phylogenies were reported by Matsuda [6] and Lewis [7], in 1995 and 1998, respectively. Following this line, other researchers published new approaches for tackling the phylogenetic inference problem. We can highlight the work of Lemmon and Milinkovitch, who published a multipopulation genetic algorithm for maximum likelihood reconstruction [8], and Congdon, who developed an evolutionary algorithm for maximum parsimony analyses [9]. Recently, the basis of bioinspired computing can be found in some

methodologies included in popular biological methods [10], [11]. One of most important questions that arises when developing these strategies is how to represent individuals in the population. Cotta and Moscato studied several direct and indirect representations, observing different advantages and disadvantages [12]. On the other hand, Poladian proposed in [13] the use of distance matrices and the Neighbour-Joining method as a genotype-phenotype mapping, applied to maximum likelihood. His proposal achieved promising results with regard to other popular heuristic-based approaches.

Recent research trends suggest the use of multiobjective optimization techniques applied to Phylogenetics. This line was defined to overcome the difficulties that arise when using these previous approaches, as several sources of evidence and different optimality criteria can give as a result conflicting tree topologies. In 2006, Poladian and Jermiin developed the first multiobjective algorithm applied to phylogenetic reconstruction [14]. Afterwards, an immune-inspired multiobjective proposal for inferring phylogenies by the minimal evolution and mean-squared error criteria was proposed by Coelho et al. [15]. Finally, Cancino and Delbem published a multiobjective genetic algorithm for maximum parsimony and maximum likelihood reconstruction, PhyloMOEA [16].

Following this last line of research, in this paper we introduce a new multiobjective bioinspired approach for inferring phylogenies that represent a consensus between maximum parsimony and maximum likelihood.

3 Approaches for Inferring Phylogenies

In this section we introduce the basis of different phylogenetic methods whose characteristics will be considered in our proposal: distance methods, maximum parsimony and maximum likelihood approaches.

3.1 Distance-Based Methods

Distance-based methods [3] were proposed with the aim of inferring phylogenetic trees by processing some distance measures among species. Despite their simplicity, these methods are very popular due to the low amount of biological information lost when modelling the evolutionary process [3]. Furthermore, these approaches can lay the foundations for more complex phylogenetic searches. Distance methods generate a symmetric matrix M of $N \times N$ dimensions, where N is the number of species in input data. Given two species i and j, $M[i, j]$ contains the evolutionary distance between them. A distance measure can be computed in several ways, such as by considering the number of different characteristics found in molecular sequences, or by using statistical methods [13].

These approaches process M to generate a phylogenetic topology $T = (V, E)$, where V represents the nodes in the tree, and E contains branches modelling ancestral relationships, as well as evolutionary distances between related organisms, given by branch length values. Figure 1 shows an example of distance-based phylogenetic reconstruction. Among the variety of distance methods that can be

	Seal	Whale	Rat	Mouse	Human	Turkey	Frog	Minnow	Carp	Bull
Seal	0	0.20	0.30	0.31	0.49	0.45	0.40	0.50	0.49	0.19
Whale	0.20	0	0.24	0.25	0.43	0.39	0.34	0.44	0.43	0.19
Rat	0.30	0.24	0	0.15	0.45	0.41	0.36	0.46	0.45	0.29
Mouse	0.31	0.25	0.15	0	0.46	0.42	0.37	0.47	0.46	0.30
Human	0.49	0.43	0.45	0.46	0	0.34	0.43	0.53	0.52	0.48
Turkey	0.45	0.39	0.41	0.42	0.34	0	0.39	0.49	0.48	0.44
Frog	0.40	0.34	0.36	0.37	0.43	0.39	0	0.30	0.29	0.39
Minnow	0.50	0.44	0.46	0.47	0.53	0.49	0.30	0	0.21	0.49
Carp	0.49	0.43	0.45	0.46	0.52	0.48	0.29	0.21	0	0.48
Bull	0.19	0.19	0.29	0.30	0.48	0.44	0.39	0.49	0.48	0

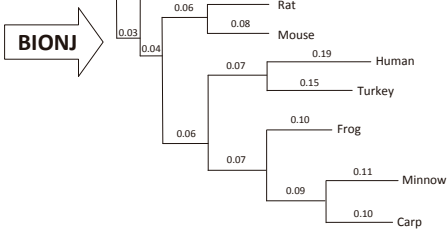


Fig. 1. An example of the BIONJ algorithm considering ten species

found in the literature, Neighbour-Joining (NJ) [17] is commonly used as an understandable way to introduce researchers into this methodology.

NJ verifies M iteratively, selecting the pair (s_1, s_2) of species that represent the closest neighbours according to the distance measures. From s_1 and s_2, the method infers their common ancestor c and includes (c, s_1, s_2) in a partial phylogeny. Entries related to s_1 and s_2 in M are replaced by c and distance values are updated. These steps are repeated until all entries in M have been processed and a complete phylogenetic topology has been inferred. In this work, we will use the BIONJ method, an extension to NJ developed by Gascuel. This proposal improves NJ performance by considering a model of variances and covariances estimated from evolutionary distances. A detailed explanation of this algorithm can be found in [18].

3.2 Maximum Parsimony Approaches

Ockham's razor has inspired a wide variety of approaches to resolve optimization problems. Maximum parsimony methods try to apply this well-known principle to Phylogenetics. Parsimony-based approaches are defined to infer those evolutionary histories that minimize the amount of molecular changes needed to explain the observed data. Given a phylogenetic tree $T = (V, E)$ that describes an evolutionary history from a set of N nucleotide sequences composed by K sites, we compute the parsimony value of T by using the following equation [19]:

$$P(T) = \sum_{i=1}^{K} \sum_{(a,b)\in E} C(a_i, b_i) \tag{1}$$

where $(a, b) \in E$ represents an ancestor-descendant relationship between the nodes a and b, a_i and b_i the states corresponding to the ith site on molecular sequences for a and b, and $C(a_i, b_i)$ the cost of evolving from the state a_i to b_i.

Those trees that minimize Equation 1 will be preferred as they imply a simpler hypothesis to the evolution of the input species. In this work we will consider

Fitch's proposal to assign ancestral sequences to internal nodes and evaluate trees according to the maximum parsimony criterion [20].

3.3 Maximum Likelihood Approaches

Likelihood-based approaches to Phylogenetics were proposed to infer the most likely evolutionary history of the organisms under review by using complex evolutionary models. Evolutionary models provide substitution matrices that define mutation probabilities at nucleotide level. By considering such models, the topology of the phylogenetic tree and branch lengths values, these methods compute statistically consistent phylogenies, according to the likelihood measurement.

Let $T = (V, E)$ be a phylogenetic tree, m an evolutionary model, and D a set of K-site nucleotide sequences representing the observed data. We can calculate the likelihood of T as follows [3]:

$$L[D, T, m] = \Pr[D|T, m] = \prod_{i=1}^{K} \prod_{j=1}^{E} (r_i t_j)^{n_{ij}} \tag{2}$$

where r_i is defined as the mutation probability for the ith site, t_j as evolutionary times given by branch $j \in E$, and n_{ij} as the number of state changes that can be found on site i between the nodes related by j.

These approaches search for those phylogenies that maximize the likelihood function, due to the fact that maximum likelihood topologies would represent the most likely evolutionary hypotheses. We will compute likelihood values by using the Felsenstein proposal [21] under the $HKY85 + \Gamma$ model [3].

4 Multiobjective Firefly Algorithm

Recently, a novel swarm intelligence algorithm inspired by the bioluminescence of fireflies was proposed by Yang [5]. The Firefly Algorithm uses concepts like brightness and attractiveness to resolve optimization problems by using collective intelligence. Fireflies behaviour is governed by a communication system based on flashing lights which allows them to attract other fireflies and to warn predators about their toxicity. Attractiveness depends on the light intensity, the distance between fireflies and the light absorption by environment. Brighter fireflies will attract less bright fireflies to their position. FA models this behaviour by considering that the light intensity of a firefly will depend on the quality of its related solution. Brighter fireflies will be associated to better solutions to the problem, so firefly population will move towards high-quality solutions.

In this study, we propose to introduce multiobjective optimization techniques to FA. For this purpose, we need to distinguish brighter fireflies to less bright fireflies in this new context. To resolve this issue, we apply *dominance*. Given two solutions x and y, we state that x dominates y if and only if x has better or equal scores than y in all objective functions and, at least, x is better in one of them. In this way, given two fireflies u and v with solutions X_u and X_v, respectively, u will be brighter than v if and only if X_u dominates X_v.

In order to adapt this Multiobjective Firefly Algorithm to phylogenetic inference, we will use the distance-based methodology proposed by Poladian in [13]. We will introduce distance matrices to model the attraction process and apply BIONJ to reconstruct the resulting phylogenetic trees. Algorithm 1 shows MO-FA pseudocode. This algorithm takes as input the following parameters:

1. swarmSize. Population size.
2. maxGenerations. Number of generations.
3. β_0. Attractiveness factor.
4. γ. Environment absorption coefficient.
5. α. Randomization factor.

Algorithm 1. MO-FA Pseudocode

```
1: X ← initializeAndEvaluatePopulation(swarmSize,dataset)
2: ParetoFront ← 0
3: i ← 0
4: while i < maxGenerations do
5:    for j = 1 to swarmSize do
6:       for k = 1 to swarmSize do
7:          /* If X[k] dominates X[j], X[j] will move towards X[k] */
8:          if X[k] ≻ X[j] then
9:             /* Compute distance from X[k] to X[j] and apply attraction formula */
10:            r_jk ← ||X[j] − X[k]|| = sqrt(Σ_{n=1}^{N} Σ_{m=1}^{n} (X[j].M[n,m] − X[k].M[n,m])^2)
11:            for each position m,n (n < m) in X[j] distance matrix do
12:               X[j].M[m,n] ← X[j].M[m,n] + β0 e^{−γ r_jk^2} (X[k].M[m,n] − X[j].M[m,n]) +
                  α(rand[0,1] − 1/2)
13:            end for
14:            X[j].T ← computeBIONJ(X[j].M)
15:            X[j] ← setParsimonyAndLikelihoodScores(X[j].T, dataset)
16:         end if
17:      end for
18:   end for
19:   X ← optimizeExtremeSolutions(X)
20:   ParetoFront ← saveSolutions(X, ParetoFront)
21:   i ← i + 1
22: end while
```

Initializing the Swarm. Each firefly in the swarm will be related to a distance matrix and the corresponding phylogenetic topology. We have used the matrix and tree templates provided by the C++ libraries for Bioinformatics, BIO++ [22]. Initial trees are selected from a repository of 1000 phylogenetic topologies generated by bootstrap techniques, 500 of them by using maximum parsimony, and the remaining 500 by maximum likelihood. In addition to this, BIO++ is used to configure the parameters of the evolutionary model.

From these topologies, initial scores and distance matrices are computed and assigned to individuals in the population. BIONJ will be used to generate phylogenetic topologies from the updated matrices during the course of the algorithm.

MO-FA Main Loop. After the firefly population has been initialized, MO-FA main loop takes place (lines 4-22 in Algorithm 1). Each dominated firefly will be modified according to the brightness and attractiveness system. For this purpose, entries in the distance matrix will be updated in order to move dominated fireflies towards the brightest ones. Firstly, given two fireflies u, v with solutions X_u and X_v, if X_u is dominated by X_v, we compute the distance r_{uv} between u and v as follows (line 10):

$$r_{uv} \leftarrow \|X_u - X_v\| = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{i}(X_u.M[i,j] - X_v.M[i,j])^2} \tag{3}$$

where $M[i,j]$ denotes the distance between species i and j. According to the analyzed dataset, distance values between fireflies can be significantly different, so we normalize resulting distances to a specific range, [0,10].

In second place, we use r_{uv} to compute the new distance matrix, updating each entry $M[i,j]$ according to MO-FA movement formula. Given the attractiveness β_0, the environment absorption coefficient γ and a randomization factor α, the updated distance between two species i and j is given by (line 12):

$$\begin{aligned} X_u.M[i,j] =& X_u.M[i,j] + \beta_0 e^{-\gamma r_{uv}^2}(X_v.M[i,j] - X_u.M[i,j]) \\ &+ \alpha(rand[0,1] - \frac{1}{2}) \end{aligned} \tag{4}$$

The second term in Equation 4 denotes how $X_u.M[i,j]$ will move towards $X_v.M[i,j]$, taking into account β_0 and γ. The third term introduces a randomization factor to the movement of fireflies which helps to maintain the population diversity, where *rand* represents a random number generator. By combining the knowledge provided by the swarm with randomness, we can address the search for quality phylogenetic topologies in undiscovered regions of the tree space.

Once the distance matrix has been updated, the BIONJ algorithm generates the new phylogenetic tree according to the new distances computed by using Equation 4. Resulting topologies will be evaluated then according to the maximum parsimony and maximum likelihood criteria (line 15).

Final Steps. After all fireflies have been processed, we apply an optimization step in order to introduce additional knowledge provided by well-known heuristic-based searches. For this reason, extreme points in Pareto front will be optimized by applying a local search procedure based on the Parametric Progressive Tree Neighbourhood (PPN) proposed by Goëffon et al. [19]. PPN neighbours will be evaluated attending to the dominance concept, with the aim of improving maximum parsimony and maximum likelihood scores. Additionally, a gradient method is applied to improve branch length values. Fireflies will learn from these new solutions, allowing the swarm to improve the quality of Pareto solutions.

At the end of the current generation, the Pareto nondominated set is updated with the current best phylogenetic trees, and a new generation takes place.

After *maxGenerations*, the Pareto set will be composed by those phylogenetic trees that suppose a compromise between the parsimony and likelihood principles.

5 Experimental Methodology and Results

In this section we explain the experimental methodology we have followed to assess the performance of the proposal, evaluating and comparing our biological results with different approaches for inferring phylogenies. We have used a well-known quality indicator in multiobjective optimization, the hypervolume metrics [2], to evaluate the quality of the inferred Pareto solutions. Hypervolume defines the size of the search space covered by our solutions, bounded by two reference points (ideal and nadir). Metaheuristics that maximize hypervolume will be preferred over other proposals in a multiobjective context. In Table 1, we show the reference points we have used to compute hypervolume values.

Table 1. Hypervolume metrics. Reference points

	Ideal Point		Nadir Point	
Dataset	Parsimony	Likelihood	Parsimony	Likelihood
*rbcL*_55	4774	-21569.69	5279	-23551.42
*mtDNA*_186	2376	-39272.20	2656	-43923.99
*RDPII*_218	40658	-132739.90	45841	-147224.59
*ZILLA*_500	15893	-79798.03	17588	-87876.39

In order to configure our proposal, we have considered a variety of values for the three main parameters of the algorithm, β_0, γ and α. The remaining parameters, *maxGenerations* and *swarmSize*, have been configured taking into account additional experiments and other authors' proposals [16]. The different values we have studied for β_0, γ and α can be found in Table 2. We have chosen by experimentation the configuration that allows MO-FA to maximize hypervolume values. Table 3 shows the resulting values for input parameters.

Table 2. Configuring parameters

Parameter	Values
β_0	{0.05, 0.1, 0.25, 0.5, 0.75, 1}
γ	{0.05, 0.1, 0.25, 0.5, 0.75, 1}
α	{0.05, 0.1, 0.25, 0.5, 0.75, 0.9}

Table 3. MO-FA input parameters

Parameter	Final value
maxGenerations	100
swarmSize	100
β_0	1
γ	0.5
α	0.05

Experiments have been carried out on four nucleotide data sets [16] using the $HKY85+\Gamma$ model: *rbcL*_55, 55 sequences of 1314 nucleotides per sequence of the rbcL gene from green plants, *mtDNA*_186, 186 sequences of 16608 nucleotides per sequence from human mitochondrial DNA, *RDPII*_218, 218 sequences of

Table 4. Experimental results

	Pareto trees	Maximum parsimony tree		Maximum likelihood tree		Best hypervolume tree		Hypervolume metrics	
Dataset		Pars.	Like.	Pars.	Like.	Pars.	Like.	Mean	Std. Dev.
rbcL_55	8	4874	-21849.36	4892	-21819.04	4882	-21830.76	70.06%	0.06428
mtDNA_186	12	2431	-39961.98	2448	-39888.58	2439	-39903.13	69.67%	0.01251
RDPII_218	39	41488	-136340.73	42833	-134169.03	41745	-135409.63	74.01%	0.33689
ZILLA_500	28	16218	-81613.47	16309	-80966.58	16221	-81212.39	69.06%	0.05880

4182 nucleotides per sequence from prokaryotic RNA, and *ZILLA_500*, 500 sequences of 759 nucleotides per sequence from rbcL plastid gene.

For each dataset, we have performed 30 independent analyses to assess the statistical relevance of the proposal. In Table 4, we summarize the results corresponding to the execution which achieved the closest score to the mean hypervolume value. Columns 3-4 and 5-6 show parsimony and likelihood values for the extreme points in Pareto front. Additionally, parsimony and likelihood scores for the non-extreme solution that contributed most to the overall hypervolume are given by Columns 7-8. Finally, mean hypervolume values and standard deviations are indicated in Columns 9-10. According to this table, the hypervolume metrics suggest that MO-FA gets significant Pareto solutions for all data sets, covering over 69% of the space bounded by reference points. Pareto fronts for each dataset can be found in Figure 2.

5.1 Comparisons with Other Proposals

In order to assess the quality of the inferred phylogenetic trees, in this subsection we compare MO-FA with other authors' multiobjective metaheuristics and popular biological methods for inferring phylogenies.

In first place, we introduce in Table 5 a comparison with PhyloMOEA, a multiobjective algorithm for maximum parsimony and maximum likelihood phylogenetic reconstruction. In this table, we show parsimony and likelihood scores for our maximum parsimony and maximum likelihood trees and compare them with the best values reported by Cancino and Delbem's proposal in [16], using $HKY85 + \Gamma$. Results suggest a significant improvement with regard to PhyloMOEA in all data sets, inferring phylogenetic trees that overcome the best scores provided by other authors' multiobjective approaches. As swarm intelligence allows the inference process to take into account knowledge provided by different fireflies, a better exploration of the tree space can be performed, dominating the results achieved by classical multiobjective evolutionary algorithms.

Secondly, we compare MO-FA with two well-known single-criterion methods from the state-of-the-art: TNT [10], for maximum parsimony reconstruction, and RAxML [11], for maximum likelihood. In Table 6 we can find the best parsimony scores achieved by MO-FA and TNT, as well as parsimony and likelihood scores for the maximum likelihood trees inferred by MO-FA and RAxML. Attending to parsimony, our proposal achieves the reference scores provided by TNT. With regard to likelihood comparison, as new versions of RAxML does not include

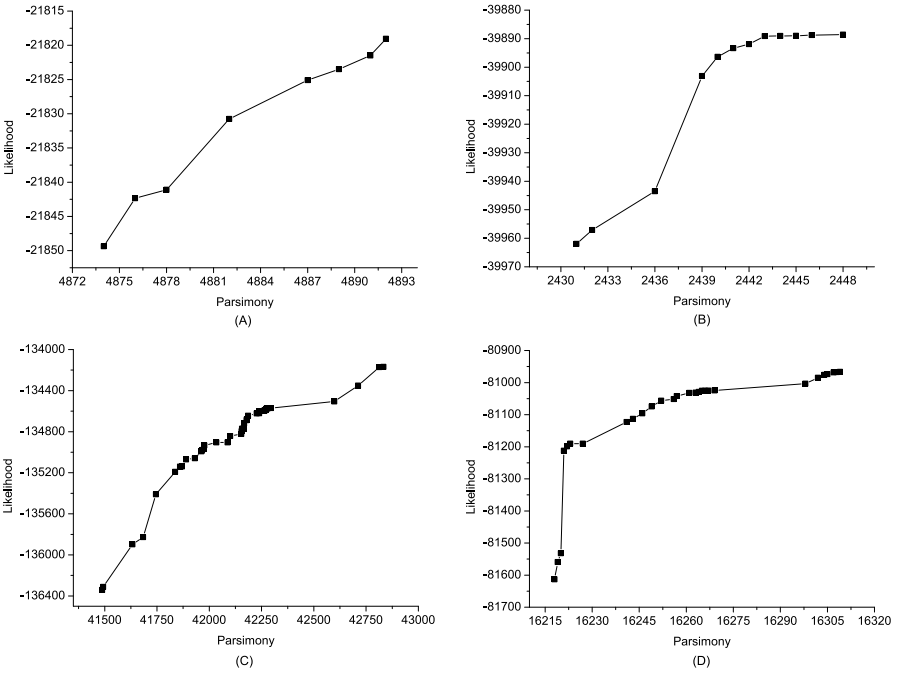


Fig. 2. Pareto fronts for *rbcL_55*(A), *mtDNA_186*(B), *RDPII_218*(C) and *ZILLA_500*(D)

Table 5. Comparing MO-FA with PhyloMOEA

	MO-FA			
	Best parsimony tree		Best likelihood tree	
Dataset	Parsimony	Likelihood	Parsimony	Likelihood
rbcL_55	**4874**	-21849.36	4892	**-21819.04**
mtDNA_186	**2431**	-39961.98	2448	**-39888.58**
RDPII_218	**41488**	-136340.73	42833	**-134169.03**
ZILLA_500	**16218**	-81613.47	16309	**-80966.58**
	PhyloMOEA			
Dataset	Best parsimony score		Best likelihood score	
rbcL_55	**4874**		-21889.84	
mtDNA_186	2437		-39896.44	
RDPII_218	41534		-134696.53	
ZILLA_500	16219		-81018.06	

Table 6. Comparing MO-FA with TNT and RAxML

	MO-FA		
	Best parsimony score	Best likelihood tree	
Dataset	Parsimony	Parsimony	Likelihood
rbcL_55	**4874**	**4890**	**-21789.27**
mtDNA_186	**2431**	**2451**	**-39869.29**
RDPII_218	**41488**	**42813**	-134089.91
ZILLA_500	**16218**	**16305**	**-80610.86**
	TNT	RAxML	
Dataset	Parsimony	Parsimony	Likelihood
rbcL_55	**4874**	4893	-21791.98
mtDNA_186	**2431**	2453	-39869.63
RDPII_218	**41488**	42894	**-134079.42**
ZILLA_500	**16218**	**16305**	-80623.50

$HKY85 + \Gamma$, we have carried out new experiments using the $GTR + \Gamma$ evolutionary model. Under this model, our likelihood topologies dominate RAxML's trees for *rbcL_55*, *mtDNA_186* and *ZILLA_500*, and improve significantly the parsimony value for *RDPII_218*. Therefore, we can suggest that a multiobjective swarm intelligence scheme allows us to obtain a meaningful performance in comparison with two of the most powerful tools for phylogenetic inference.

6 Conclusions and Future Work

We have introduced in this paper a multiobjective approach based on the collective behaviour of fireflies for tackling the phylogenetic inference problem according to two well-known criteria: maximum parsimony and maximum likelihood. In order to model fireflies' behaviour, we have used a distance-based methodology supported by the BIONJ algorithm, where distance matrices are computed and processed to generate new phylogenetic topologies. Experiments on four public nucleotide data sets show that this swarm intelligence proposal can achieve significant performance in comparison with other multiobjective evolutionary algorithms and state-of-the-art biological methods, inferring a set of trade-off phylogenetic trees by considering the parsimony and likelihood principles.

As future work, we will introduce this distance-based methodology and individual representation into a previous swarm intelligence algorithm for inferring phylogenies, Multiobjective Artificial Bee Colony (MOABC) [23], with the aim of making possible a fair comparison between MOABC and MO-FA. The reason why we need to study such step is because performing this comparison without taking into account the same experimental conditions can give as a result biased conclusions. Additionally, other distance methods besides BIONJ will be studied, in order to assess which one can lead MO-FA to improved performances. Finally, we will apply parallel computing to improve the efficiency of the proposal by exploiting modern hardware architectures.

Acknowledgment. This work was partially funded by the Spanish Ministry of Economy and Competitiveness and the ERDF (European Regional Development Fund), under the contract TIN2012-30685 (BIO project). Thanks to the Fundación Valhondo Calaff for the financial support offered to Sergio Santander-Jiménez.

References

1. 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)
2. Zitzler, E., Thiele, L.: Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. IEEE Transactions on Evolutionary Computation 3(4), 257–271 (1999)
3. Felsenstein, J.: Inferring phylogenies. Sinauer Associates, Sunderland (2004) ISBN: 0-87893-177-5
4. Wiens, J.J., Servedio, M.R.: Phylogenetic analysis and intraspecific variation: performance of parsimony, likelihood, and distance methods. Systematic Biology 47(2), 228–253 (1998)
5. Yang, X.-S.: Firefly Algorithm, Stochastic Test Functions and Design Optimisation. Int. J. Bio-Inspired Computation 2(2), 78–84 (2010)
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. Mol. Biol. Evol. 15(3), 277–283 (1998)
8. Lemmon, A.R., Milinkovitch, M.C.: The metapopulation genetic algorithm: An efficient solution for the problem of large phylogeny estimation. Proceedings of the National Academy of Sciences USA 99(16), 10516–10521 (2002)
9. Congdon, C.: GAPHYL: An evolutionary algorithms approach for the study of natural evolution. In: Genetic and Evolutionary Computation Conference, GECCO 2002, pp. 1057–1064 (2002)
10. Goloboff, P.A., Farris, J.S., Nixon, K.C.: TNT, a free program for phylogenetic analysis. Cladistics 24, 774–786 (2008)
11. Stamatakis, A.: RAxML-VI-HPC: Maximum Likelihood-based Phylogenetic Analyses with Thousands of Taxa and Mixed Models. Bioinformatics 22(21), 2688–2690 (2006)
12. Cotta, C., Moscato, P.: Inferring Phylogenetic Trees Using Evolutionary Algorithms. In: Guervós, J.J.M., Adamidis, P., Beyer, H.-G., Fernández-Villacañas, J.-L., Schwefel, H.-P. (eds.) PPSN VII. LNCS, vol. 2439, pp. 720–729. Springer, Heidelberg (2002)
13. Poladian, L.: A GA for Maximum Likelihood Phylogenetic Inference using Neighbour-Joining as a Genotype to Phenotype Mapping. In: Genetic and Evolutionary Computation Conference, GECCO 2005, pp. 415–422 (2005)
14. Poladian, L., Jermiin, L.: Multi-Objective Evolutionary Algorithms and Phylogenetic Inference with Multiple Data Sets. Soft Computing 10(4), 359–368 (2006)
15. 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)
16. 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
17. Saitou, N., Nei, M.: The Neighbor-joining Method: A New Method for Reconstructing Phylogenetic Trees. Molecular Biology and Evolution 4(4), 406–425 (1987)
18. Gascuel, O.: BIONJ: An Improved Version of the NJ Algorithm Based on a Simple Model of Sequence Data. Molecular Biology and Evolution 14(7), 685–695 (1997)
19. Goëffon, A., Richer, J.M., Hao, J.K.: Progressive Tree Neighborhood Applied to the Maximum Parsimony Problem. IEEE/ACM Transactions on Computational Biology and Bioinformatics 5, 136–145 (2008)
20. Fitch, W.: Toward Defining the Course of Evolution: Minimum Change for a Specific Tree Topology. Systematic Zoology 20(4), 406–416 (1972)
21. Felsenstein, J.: Evolutionary Trees from DNA Sequences: A Maximum Likelihood Approach. Journal of Molecular Evolution 17, 368–376 (1981)
22. 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, 188–193 (2006)
23. Santander-Jiménez, S., Vega-Rodríguez, M.A., Gómez-Pulido, J.A., Sánchez-Pérez, J.M.: Inferring Phylogenetic Trees Using a Multiobjective Artificial Bee Colony Algorithm. In: Giacobini, M., Vanneschi, L., Bush, W.S. (eds.) EvoBIO 2012. LNCS, vol. 7246, pp. 144–155. Springer, Heidelberg (2012)

Mining for Variability in the Coagulation Pathway: A Systems Biology Approach

Davide Castaldi[1], Daniele Maccagnola[1], Daniela Mari[2,3], and Francesco Archetti[1,4]

[1] DISCo, University of Milan-Bicocca, Viale Sarca 336, Milan, 20126, Italy
[2] IRCCS Cà Granda Ospedale Maggiore Policlinico Foundation, Via Pace 9, Milan, 20122, Italy
[3] Department of Clinical Sciences and Community Health, University of Milan, 20122 Italy
[4] Consorzio Milano Ricerche, Via Cozzi 53, Milan, 20126, Italy
{davidefabio.castaldi,daniele.maccagnola}@disco.unimib.it,
daniela.mari@unimi.it, francesco.archetti@unimib.it

Abstract. In this paper authors perform a variability analysis of a Stochastic Petri Net (SPN) model of the Tissue Factor induced coagulation cascade, one of the most complex biochemical networks. This pathway has been widely analyzed in literature mostly with ordinary differential equations, outlining the general behaviour but without pointing out the intrinsic variability of the system. The SPN formalism can introduce uncertainty to capture this variability and, through computer simulation allows to generate analyzable time series, over a broad range of conditions, to characterize the trend of the main system molecules. We provide a useful tool for the development and management of several observational studies, potentially customizable for each patient. The SPN has been simulated using Tau-Leaping Stochastic Simulation Algorithm, and in order to simulate a large number of models, to test different scenarios, we perform them using High Performance Computing. We analyze different settings for model representing the cases of "healthy" and different "unhealthy" subjects, comparing and testing their variability in order to gain valuable biological insights.

Keywords: Systems Biology, Variability Analysis, Coagulation, Stochastic Simulation, Petri Nets.

1 Introduction

Growing evidences of modelling biological systems require taking into account "uncertainty" in system behaviour, due to external interferences (drugs, environment, etc), inter-variability (personal features) and intra-variability (intrinsic noise) of complex systems. Moreover, the simulation of biochemical reactions has allowed, during the years, to acquire relevant information about specific behaviours of species involved in the processes. A better understanding of the system means to denote molecular relational aspects that could be hardly to evaluate in wet laboratory, due to the complexity of assays for investigate them. Petri Net formalism is a modeling tool widely used to represent biological systems. In contrast to other modelling tools for

M. Middendorf and C. Blum (Eds.): EvoCOP 2013, LNCS 7832, pp. 153–164, 2013.

biological applications, Petri Nets offer a more formal view and the underlying theory is well understood. The graphical aspects of the Petri Net are quite similar to biochemical network representation, and this gives superior communication ability to models and facilitates their design, in particular for complex structure. Their stochastic extension, Stochastic Petri Nets (SPNs), are effective for modeling the dynamics of stochastic biological process, in order to assess the peculiar changeability of such systems [11] and allow to evaluate hidden molecular behaviours [14].

This model, in the previous decade, has been deeply investigated, resulting in complex networks where two sub-pathways, *Intrinsic* and *Extrinsic* interact in a *Common* pathway to produce active fibrin which prompts the clot formation.

The existing computational models of the coagulation cascade are based on deterministic approaches [13], but a few works have tried to add stochasticity to describe the process [7]. However, none of these stochastic approaches allows a detailed analysis of the coagulation system and thrombin generation var, whose results can be representing existing clinical tests, in particular the most widely used test for monitoring oral anticoagulation therapy, the "Prothrombin Time" (PT) test.

In this paper the authors developed a model of the coagulation (Sec.2), represented by counterbalance of coagulation and anticoagulation proteins, based on a Petri Net modeling framework (Sec.3). This model offers an intuitive graphical representation of TF pathway with a manageable modularity. The presented model represents a new stochastic bioclinical approach of modelization of the extrinsic pathway, and the variability analysis, enables to give more strength and comparability to the results obtained. This work is focused on comparing the behaviour of a "healthy" subject with that of different "unhealthy" subject, starting from a Stochastic Petri Net (SPN) model and changing the initial marking to represent the effect of a prothrombotic event. Unlike other similar works, we perform a set of analysis on the computational results in order to evaluate their significance and give more strength to the results obtained. The models will be first fitted to literature data using a deterministic approach, converting the SPN into a Continuous Petri Net (CPN) and solving the related system of ODEs. The actual analysis will be performed with stochastic simulation algorithms, employing Gillespie's Stochastic Simulation Algorithm (SSA) and in particular its variant called Tau-Leaping (Sec.4). It is clear that the stochastic model is more effective than its deterministic counterpart in unearthing valuable biological insights, but its computational demands can become prohibitive even for moderate size networks. Approximate methods, along with the use of high performance computing (HPC) are shown in Sec.5 to bring the computational complexity of the stochastic approach within manageable limits.

2 Biological Case Study and Petri Nets

The currently accepted model of *in vivo* coagulation highlights the central importance of tissue factor (TF) as the main instigator of coagulation, while emphasizing the rapid amplification of thrombin as an essential step in the development of a stable clot [2]. Even this partial pathway is built as the classical cascade in which are defined stepwise and overlapping patterns of activation, with an initial, an amplification, a

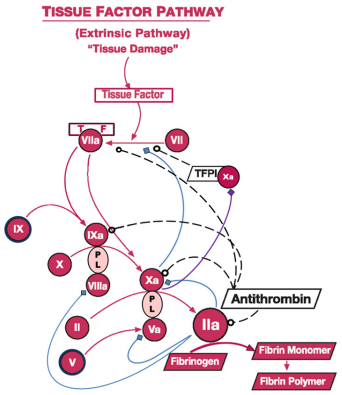
propagation, and an inhibition phases. The graphical representation of the Petri net, quite similar to a biochemical network, allows to model coagulation enzymes and cofactors as places, and the reactions which involve these elements are represented by the transitions. The arcs which connect places and transitions will represent the stoichiometry of the reaction, while the tokens associated to a place will represent the amount of molecules of that particular reactant (pre-place) or product (post-place). This paper employs notation used in [12]. In terms of a chemical reaction network, coagulation involves a sequence of highly connected concurrent processes with many simultaneous positive and negative feedback loops that specify the beginning, the progression and the amplification of whole system.

Using this formalism, we have been able to build up a model of the coagulation pathway. In particular, the relevant role played in coagulation process by the Tissue Factor [6], also in cardiovascular and cancer [16] diseases, has supported the idea of modeling the extrinsic pathway in details. We accurately reproduce the pathway, including all the positive and negative feedback circuits, as well as the exact number of molecules involved in the process.

3 Petri Net Model Construction

The initial marking represents in our model the average value of the observed physiological range The reaction rates have been tuned with a trial and error approach (based on enzyme kinetic assumptions) to replicate the thrombin generation time given by a bioclinical PT test [13], and reproduce a titration curve of the thrombin formation in a biochemical test [3]. The modularity of the system facilitated the tuning phase, as each single module has been modeled and tuned separately before merging the whole pathway in one model. To define the initial marking and reaction rate constants, we examined biological databases as Brenda (http://www. brenda-enzymes.org), model repositories as BioModel Database (http:// www.ebi.ac.uk/biomodels-main), integrated with other data found in PubMed (http://www.ncbi.nlm.nih.gov/pubmed). In the initial marking (M_0), only 11 places have a non-zero number of tokens, which ranges from 75 to $3{,}01 \cdot 10^8$ molecules (the number has been deduced from the values reported in Fig.1), with a considered plasma volume of $1 \cdot 10^{-10}$ liters. The whole set of places, transitions and parameters (initial marking and rate constants) are given as supplementary material on the authors' website (www.nedd.unimib.it - Downloads - Supplementary Materials).

Techniques for the automatic discovery of the correct parameters available in literature [18, 22] were hardly applicable to our model because of the very large number of molecules described and because of the lack of comparable raw data of previous coagulation models (which are only presented in percentual values). Fig.2 shows the TF pathway represented as a Petri Net model. We build up the PN graph using modules, based on the four-phase structure characteristic of the pathway and we used macro-nodes to give a neat and hierarchically structured representation of our model. Fig.3 shows in detail the two types of reactions described in the hierarchical PN: complex association/dissociation (AD#) and Michaelis-Menten enzyme reaction (MM#). The hierarchical structure of the model does not change any properties of the flat model [1].



Procoagulant Proteins

Species	Molec. Weight (kDa)	Plasma Level (ug/ml)
Fibrinogen (I)	340	3000
Prothrombin (II)	72	90
Factor V	330	10
Factor VII	50	0.5
Factor VIII	295	0.1
Factor IX	57	5
Factor X	59	8
Tissue Factor	45	0.001
VIIa-TF Complex	95	0.0005

Anticoagulant Proteins

Species	Molec. Weight (kDa)	Plasma Level (ug/ml)
TFPI	38	0.001
Antithrombin III	58	290

Fig. 1. The wall chart representation of Tissue Factor Pathway (modification of image from http://www.enzymeresearch.co.uk) and species physiological characteristics. The pathway can be subdivided in a procoagulant subpathway (red arcs), positive feedbacks (blue arcs), negative feedbacks (purple arcs) and inhibitor subpathways (black dashed arcs).

Our network is composed, in total, by 26 places and 18 macro-nodes, which can be unfolded into a network of 35 places and 43 transitions. The standard semantics for qualitative Petri nets does not associate a time with transitions or the sojourn of tokens at places, and thus these descriptions are time-free. The qualitative analysis considers however all possible behaviour of the time-independent system. In SPNs, the choice of the transition that will be fired is no longer deterministic. The firing rate of every transition is represented by a probability, simulating the chance that a transition will fire in a given state. The firing rates may depend on the state of the system, in particular the distribution of the tokens over the system

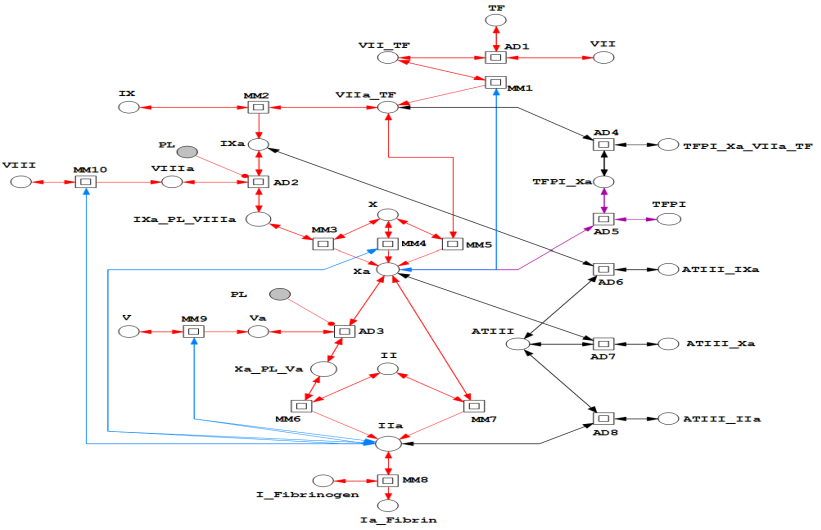


Fig. 2. High-level Petri Net model of Tissue Factor pathway

According to Goss and Peccoud [11] enabled transitions in SPNs fire with an negative exponentially distributed time delay function:

$$f_{X_t} = \lambda_t(m) \cdot e^{(-\lambda_t(m)\cdot\tau)} \quad , \tau \geq 0$$

where the waiting time X_t is function of the transition rate $\lambda_t(m)$ (a *stochastic mass-action hazard function* as defined in [12]) and the time τ.

SPN are a specification language which enables both a deterministic solver based on ODE (through another specific called Continuous Petri Nets), and a stochastic solver based on Gillespie Stochastic Simulation Algorithm (SSA). We use the deterministic approach to compare the average behaviour of our model to literature, in order to tune the parameters (in particular the reaction costants). Among the existing works, the mathematical model which comes closest to our purpose has been proposed by Khanin, Rakov and Kogan [13], in terms of similarity of the considered pathway and similarity of thrombin time production time. The details of the comparison are given in the supplementary materials. The deterministic approach is useful to compare the average behaviour of our model to literature, but it cannot highlight other important characteristics unidentifiable in the average behaviour, such as the biological systems variability. These problems will be overcome with the stochastic simulation approach.

(a) (b)

Fig. 3. Description of macro-nodes. (a) Representation of association/dissociation of a complex (AD# in high-level PN). (b) Representation of a Michaelis-Menten enzyme reaction (MM# in high-level PN).

4 Stochastic Simulation Approach

Deterministic simulation techniques assume that concentrations vary deterministically over time and that concentrations vary continuously and continually. However, these assumptions may not be valid for some important aspects of biological systems and this limitation can hamper capturing important properties of the system under study. In particular, a deterministic approach is not accurate when the species occurs in small molecular quantities, and it cannot represent the intrinsic variation of the model, which in these systems has a very important biological meaning [5]. Our approach based on Stochastic Petri Nets allows studying the stochastic behaviour of the coagulation model through the use of stochastic simulation algorithms.

The most common stochastic simulation algorithm (SSA) is the "Direct Method" proposed by Gillespie [9]; it explicitly simulates each reaction event in the system, therefore it correctly accounts for all stochastic fluctuations. Thus, the algorithm has time complexity approximately proportional to the overall number of particles and

reactions in the system. Direct Method uses two random numbers per step to compute which will be the next reaction and when it will happen (the time step τ), as described in [9]. This method has the best accuracy among all Stochastic Simulation Algorithms, but it has a prohibitive time complexity for a system with a high number of reactions and particles. In order to accelerate simulation of the coagulation model, we used a faster SSA called 'Tau-leap method' [10]. This is faster than Direct Method, because it avoids the simulation of every single reaction event. Instead, it "leaps" along the time axis in steps of length τ, which contains many single reaction events. τ must fulfill a property called "Leap Condition", which means it has to be small enough that no significant change in the reaction rates occurs during $[t, t+\tau]$. Choosing the correct value for τ is a problem much discussed in literature [10, 4]. Tau-Leap method allowed us to simulate the coagulation model much faster than the Direct Method, because when the particle number increases, many reactions are simulated at once within one time step.

5 Experimental Results

Our main work is focused on simulating different models of haemostasis, some with healthy behaviour and some with unhealthy behaviour, in order to test the behavioural changes under the influence of external stimuli. For each kind of model we perform many simulations, in order to represent the real physiological changeability of coagulation observable in laboratory tests. We can consider the phenomenon of variability of this system from two points of view, in which different simulation represent:

Inter-variability: different subjects we assume to have the same initial marking, that lead them to have different patterns of coagulation;

Intra-variability: a single subject, with an average initial marking, who can show different trends because of the intrinsic variability.

In both cases, performing several simulations we can observe how the variability of biological system is reproducible by stochastic simulation. This approach allows us to evaluate whether the sample paths generated by simulation follows the real behaviour of the system in normal and stressed conditions, allowing the validation of our models.

5.1 Modeling and Simulation Framework

In order to perform simulation on our models, we employed two different tools:

- *Snoopy* [20] has been used as a tool to build the models as Stochastic Petri Nets, to convert them to Continuous Petri Nets and perform deterministic simulation of the associated system of ODEs. Snoopy also allows to perform stochastic simulations using Direct Method SSA, but since in this work we based on Tau-Leaping Method, we chose to employ another tool for these simulations.
- *StochKit2* [21], a biochemical kinetics simulation software package, which includes a series of tools to perform stochastic simulations of the models, in particular Tau-Leaping Method SSA.

Fig. 4 shows the scheme we followed to simulate the models. The first step is building a SPN based on the notions on coagulation, then we have to find the correct parameters in order to fit the model to literature results (as described in Sec.3). This is done by generating the ODE system (based on a CPN) and solve it using deterministic solvers available in Snoopy.

Once we have found the correct parameters, we export the SPN information in a SBML document and we use StochKit2 to simulate the model. Here we have two available stochastic simulation methods: Direct Method and Tau-Leaping Method.

At first, the simulations have been performed on a Dell Studio XPS 7100 with AMD Phenom II X6 1035T 2,60 GHz (with 6 cores) and 8 GB RAM, in order to test a small number of results and the average time requirements. After fixing the parameters of both models using deterministic approach, we have analyzed the variability of the system with a stochastic method. We tested both Direct Method SSA and Tau-Leaping Method, in order to find the best algorithm in terms of accuracy and time complexity. Direct Method takes a prohibitive time because of the size of the model. A single simulation takes 350 hours to simulate 10 seconds of coagulation, and simulating the model up to 25 seconds (the minimal time required for analyze the behaviour of the model) might take many months to complete. Tau-Leaping Method allows a faster simulation, inducing only a minor loss in accuracy. It takes around 8 hours to simulate 25 seconds of coagulation, performing one simulation for each core (for a total of six simulations in 8 hours).

Our focus is on computation of an ensemble of Tau-Leaping SSA realizations, and we need at least 100 simulation results to estimate the characteristics of a model with an acceptable statistical accuracy. As our six-core system can perform only six realizations at a time, the whole process would require more than 120 hours. Thus, we decided to execute the algorithm on the Bari INFN Computer Farm (based on Torque/Maui, and composed of ~3700 CPU/Core with a disk space of ~1.3PByte), which allowed us to parallelize the process and simulate all the 100 realizations simultaneously, requiring a total of 8 to 16 hours to complete them (depending on the model and on the clock rate of the available cores). The simulations were performed on a single cluster, using up to 100 nodes to simulate all the realization simultaneously.

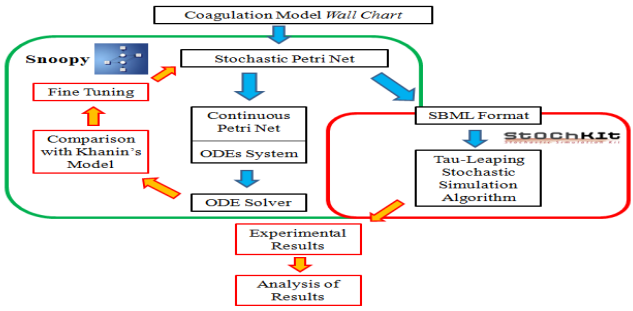
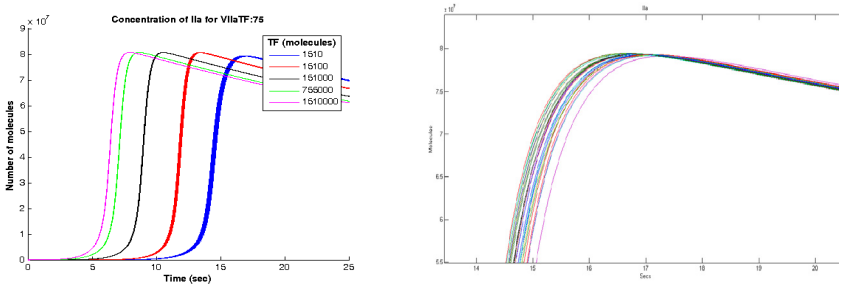


Fig. 4. Scheme of the simulation framework

We computed the trajectories and significant values such as peak times (in particular, their mean and standard deviation). It is important to note that we parallelize only the ensemble of SSA realizations, but each single simulation is not split on multiple processors. Most of the attempts in literature parallelized across the simulations, trying to speed up the process using GPUs [15] and improving the random number generation. Attempts have been made to parallelize single simulations (e.g. see [8]), but these algorithms work by splitting the set of reactions among blocks, and the structure of our Stochastic Petri Net cannot be easily partitioned because of the strong interconnections among the nodes.

5.2 Models Description

In healthy physiological conditions constant generation of small amounts of coagulation active factors (active proteases) affect the constitutive heamostasis, while the inhibitor mechanism counterweighs the process. Together, coagulation, natural anticoagulation define a delicate physiological balance; significant deviations from this equilibrium, hypercoagulability or hypocoagulability unbalance, can evolve in cardiovascular adverse events. This work is than focused on comparing the behaviour of a "healthy" subject with that of an "unhealthy" subject. We generate different models starting from the same Stochastic Petri Net shown in Sec.3, but changing the initial marking to represent the effect of a prothrombotic event. In particular, the unhealthy models have a higher initial number (from 10-fold to 1000-fold) of molecules in place representing the trigger factor of the extrinsic pathway, the Tissue Factor (TF place).



(a) (b)

Fig. 5. (a) Thrombin (IIa) molecules trend in healthy and unhealthy models; (b) Enlargement of the healthy model trends (blue curves)

This condition is clinically supported, reflecting the hypercoagulability state that arises locally during atherosclerotic plaque rupture [19]. A second factor influencing the system is the VIIaTF complex which, as highlighted in literature [17], plays an important role in the thrombus formation process. A minimal amount is needed to start the coagulation cascade, but an excess of this complex due to prior cardiovascular inflammatory events can significantly affect the coagulation process. Therefore,

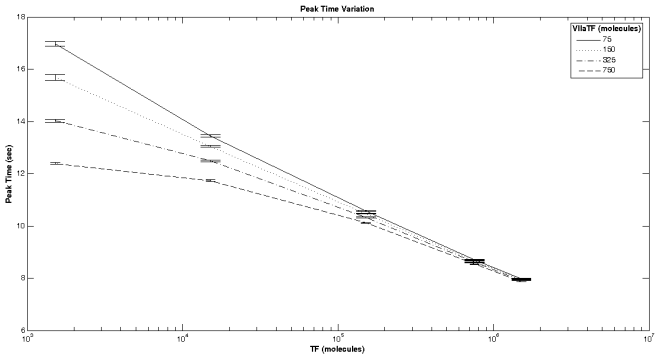


Fig. 6. Variation in mean and standard deviation of Thrombin (IIa) peak time

we test the behaviour of our model with different amount of this factor, comparing a physiological amount with a pathological one, represented by a 2-fold, 5-fold and a 10-fold increase. It is important to note that the reaction rate constants has not been modified from healthy to unhealthy models, because biological evidence proves that enzyme kinetics does not vary; nonetheless, the reaction firing rates change considerably between the two models because of the initial marking, promoting procoagulatory events.

5.3 Analysis of Computational Results

Data generated by the multiple settings of the model have been collected in a single dataset, which has been subsequently analyzed using data mining tools. The dataset collects in particular the value of the peak time of each sample path generated by the model, for a total of 100 time series for each of the 20 settings (5 variations for TF and 4 for VIIaTF).

Table 1. Mean standard deviation and *p* of peak value and peak time, in healthy and different unhealthy models

	VIIaTF Normal		VIIaTF 2-Fold		VIIaTF 5-Fold		VIIaTF 10-Fold	
Model	**Thrombin Mean Peak Time (sec)**	**Thrombin St.Dev. Peak Time (sec)**	**Thrombin Mean Peak Time (sec)**	**Thrombin St.Dev. Peak Time (sec)**	**Thrombin Mean Peak Time (sec)**	**Thrombin St.Dev. Peak Time (sec)**	**Thrombin Mean Peak Time (sec)**	**Thrombin St.Dev. Peak Time (sec)**
Healthy	16,96	0,106	15,67	0,091	14,03	0,054	12,38	0,037
Un.(TF 10-fold)	13,44	0,051	13,04	0,044	12,49	0,039	11,73	0,026
Un.(TF 100-fold)	10,56	0,017	10,46	0,017	10,33	0,016	10,11	0,015
Un.(TF 500-fold)	8,92	0,012	8,66	0,011	8,61	0,009	8,51	0,009
Un.(TF 1000-fold)	7,97	0,008	7,94	0,008	7,91	0,008	7,86	0,008

As we can see from Fig.5a, the amount of generated thrombin does not change from the healthy model to the unhealthy one, because the total availability of its precursor (prothrombin, II) is the same. Otherwise, we can see an anticipation of the growth and peak production in the unhealthy model, which is due only to the change of initial condition. This is observable even with a deterministic approach, but only from the stochastic results (see Table 1 and Fig.6) we can notice how the unhealthy models show a lower intrapersonal variability compared to the healthy model, which shows a higher variability. A likely explanation of the lower variability in the unhealthy subject is given by the augmented amount of TF molecules, which leads to a higher rate for the procoagulant reactions. Since the rate of the inhibitory reactions is less affected, they will fire with a lower frequency, thus reducing the noise on the main cascade. This is consistent with the clinical evidence, where the biological variability is lower in patients with prothrombotic conditions, and higher in patients with pro-haemorragic phenotypes. The same effect is produced by an increased amount of factor VIIaTF, which also limits the influence of the increment of TF amount (high levels of TF do not affect a system with high levels of VIIaTF). We apply a set of analysis on the computational results in order to evaluate their significance and give more strength to the results obtained. We perform an exploratory data analysis on distribution and variance to establish which tests were more appropriate. Biological data are often lognormal, with unequal variances when the means differ. Hence, we evaluate the normal distribution of all data through a one-sample Kolgomorov-Smirnov test. From a comparison of different ANOVA tests, we decided to perform the Levene-median equality of variances test, which ensures the robustness of the evaluation for non-normal data and equal sample size (the tests have been performed using PASW Modeller (IBM SPSS software)). The Levene test results (see Table 2) confirms that these variances are significantly different, which is particularly important given that the peaks of curves represent the resulting time of bioclinical PT test. This confirms that an increase of Tissue Factor appreciably reduces the changeability of the coagulation cascade, and also that an increase of VIIaTF complex reduces this effect.

Table 2. Levene Equality of Variances test for incremental TF amounts. The variances are significantly different in all the cases, but the significance level decreases for higher values of VIIaTF.

VIIa_TF Value	Levene *F*-Statistic	d.f.	Sig.
Normal	33,011	3	1.68E-9
2-Fold	23,990	3	3.76E-5
5-Fold	21,632	3	4.31E-4
10-Fold	19,856	3	2.86E-3

6 Conclusion

In this paper we modeled the extrinsic coagulation system using a Stochastic Petri Net formalism, obtaining a valuable tool for the simulation and the analysis of the

behaviour of this complex network. We employed Gillespie's Tau-Leaping SSA in order to reduce the simulation time while retaining a good approximation, and we developed models describing "healthy" and "unhealthy" subjects. The stochastic approach has been proven to detect important features that deterministic models cannot discover: in particular, we shown that an increase in the quantity of Tissue Factor or VIIaTF complex reduces the degree of variation of the system, and we evaluated the significance of this assessment through Levene statistical test. This result matches clinical evidence where the biological variability is lower in of patients with prothrombotic conditions, and higher in patients with normal phenotypes. This model can capture the true variability of this complex system as it represents the coagulation in a more realistic manner compared to the deterministic models appeared in literature. The modularity with which this model has been constructed sets the background for adding further details such as pharmacological interactions or the applicability to other clinical domains. This approach can be useful in the field of personalized medicine, where a suitable setting of the model can help to determinate in advance how much a patient response to the treatment will vary. Other developments may include the automatic stochastic simulation of models with a wide range of parameters, which would allow to detect other important components and features of the coagulation system.

References

[1] Breitling, R., Gilbert, D., Heiner, M., Orton, R.: A structured approach for the engineering of biochemical network models, illustrated for signalling pathways. Brief Bioinform. 9(5), 404–421 (2008)

[2] Bungay, S.: Modelling the effect of amplification pathway factors on thrombin generation: A comparison of hemophilias. Transfus. Apher. Sci. 38, 41–47 (2008)

[3] Butenas, S., van't Veer, C., Mann, K.G.: "Normal" Thrombin Generation. Blood 94(7), 2169–2178 (1999)

[4] Cao, Y., Gillespie, D.T., Petzold, L.R.: Efficient stepsize selection for the tau-leaping simulation method. J. Chem. Phys. 124, 044109 (2006)

[5] Cevenini, E., Bellavista, E., Tieri, P., Castellani, G., Lescai, F., Francesconi, M., Mishto, M., Santoro, A., Valensin, S., Salvioli, S., Capri, M., Zaikin, A., Monti, D., de Magalhaes, J.P., Franceschi, C.: Systems biology and longevity: an emerging approach to identify innovative anti-aging targets and strategies. Curr. Pharm. Des. 16(7), 802–813 (2010)

[6] Chu, A.J.: Tissue factor, blood coagulation, and beyond: an overview. Int. J. Inflam. 367284, 1–30 (2011)

[7] Corlan, A.D., Ross, J.: Canalization effect in the coagulation cascade and the interindividual variability of oral anticoagulant response. A simulation study. Theor. Biol. Med. Model. 8, 37 (2011)

[8] Dittamo, C., Cangelosi, D.: Optimized Parallel Implementation of Gillespie's First Reaction Method on Graphics Processing Units. In: International Conf. on Computer Modeling and Simulation (ICCMS 2009), pp. 156–161 (2009)

[9] Gillespie, D.T.: Exact Stochastic Simulation of Coupled Chemical Reactions. The Journal of Physical Chemistry 81(25), 2340–2361 (1977)

[10] Gillespie, D.T.: Approximate accelerated stochastic simulation of chemically reacting systems. J. Chem. Phys. 115, 1716–1733 (2001)
[11] Goss, P.J.E., Peccoud, J.: Quantitative modeling of stochastic systems in molecular biology by using stochastic Petri nets. Proc. Nat. Acad. Sci. USA 95, 6750–6754 (1998)
[12] Heiner, M., Gilbert, D., Donaldson, R.: Petri Nets for Systems and Synthetic Biology. In: Bernardo, M., Degano, P., Zavattaro, G. (eds.) SFM 2008. LNCS, vol. 5016, pp. 215–264. Springer, Heidelberg (2008)
[13] Khanin, M.A., Rakov, D.V., Kogan, A.E.: Mathematical model for the blood coagulation prothrombin time test. Thromb. Res. 89(5), 227–232 (1998)
[14] Levine, E., Hwa, T.: Stochastic fluctuations in metabolic pathways. PNAS 104(22), 9224–9229 (2007)
[15] Li, H., Petzold, L.: Efficient Parallelization of the Stochastic Simulation Algorithm for Chemically Reacting Systems on the Graphics Processing Unit. Int. J. of High Perf. Comp. Appl. 24, 107–116 (2010)
[16] Liu, Y., Jiang, P., Capkova, K., Xue, D., Ye, L., Sinha, S.C., Mackman, N., Janda, K.D., Liu, C.: Tissue Factor Activated Coagulation Cascade in the Tumor Microenvironment Is Critical for Tumor Progression and an Effective Target for Therapy. Cancer Res. 71, 6492–6502 (2011)
[17] Monroe, D.M., Key, N.S.: The tissue factor-factor VIIa complex: procoagulant activity, regulation, and multitasking. J. Thromb. Haemost. 5(6), 1097–1105 (2007); Review
[18] Nobile, M.S., Besozzi, D., Cazzaniga, P., Mauri, G., Pescini, D.: Estimating reaction constants in stochastic biological systems with a multi-swarm PSO running on GPUs. In: Soule, T. (ed.) Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference Companion (GECCO Companion 2012), pp. 1421–1422. ACM, New York (2012)
[19] Reininger, A.J., Bernlochner, I., Penz, S.M., Ravanat, C., Smethurst, P., Farndale, R.W., Gachet, C., Brandl, R., Siess, W.: A 2-Step Mechanism of Arterial Thrombus Formation Induced by Human Atherosclerotic Plaques. J. Am. Coll. Cardiol. 55, 1147–1158 (2010)
[20] Rohr, C., Marwan, W., Heiner, M.: Snoopy—a unifying Petri net framework to investigate biomolecular networks. Bioinformatics 26(7), 974–975 (2010)
[21] Sanft, K.R., Wu, S., Roh, M., Fu, J., Lim, R.K., Petzold, L.R.: StochKit2: software for discrete stochastic simulation of biochemical systems with events. Bioinformatics 27(17), 2457–2458 (2011)
[22] Shaw, O., Steggles, J., Wipat, A.: Automatic Parameterization of Stochastic Petri Net Models of Biological Networks. Electronic Notes in Theoretical Computer Science 151, 111–129 (2006)

Improving the Performance of CGPANN for Breast Cancer Diagnosis Using Crossover and Radial Basis Functions

Timmy Manning[1] and Paul Walsh[2]

[1] Cork Institute of Technology, Bishopstown, Cork, Ireland
timothy.manning@mycit.ie
[2] NSilico Ltd., Melbourne Building, Bishopstown, Cork, Ireland
paul.walsh@nsilico.com

Abstract. Recently published evaluations of the topology and weight evolving artificial neural network algorithm Cartesian genetic programming evolved artificial neural networks (CGPANN) have suggested it as a potentially powerful tool for bioinformatics problems. In this paper we provide an overview of the CGPANN algorithm and a brief case study of its application to the Wisconsin breast cancer diagnosis problem. Following from this, we introduce and evaluate the use of RBF kernels and crossover to CGPANN as a means of increasing performance and consistency.

Keywords: CGP, CGPANN, Wisconsin breast cancer, neuroevolution, radial basis functions, crossover.

1 Introduction

Despite the acknowledged power and success of artificial neural networks (ANNs), the identification of optimal architecture is still an unsolved task [10]. The performance and potential of a network is influenced by the number of neurons, their organization and their level of interconnection. Overly large networks tend to overfit problems at the expense of generalization, while networks with an insufficient numbers of neurons or synapses are unable to encode sufficiently complex mappings, leading to suboptimal performance [21]. Evaluating architectures can also be problematic as the typical gradient descent algorithms used to train the networks are inconsistent as they can become trapped in local minima of the error space [2].

These limitations to an otherwise powerful approach have promoted interest in neuroevolution: the application of evolutionary algorithms to identifying good ANN parameters in a methodical fashion. Although neuroevolution is not guaranteed to produce the optimal weights or architecture, the use of genetic algorithms can provide an efficient and intelligent search of the space of possible configurations to produce somewhat optimal solutions in a reasonable time frame.

L. Vanneschi, W.S. Bush, and M. Giacobini (Eds.): EvoBIO 2013, LNCS 7833, pp. 165–176, 2013.

An interesting recent development in the field of Topology and Weight Evolving Artificial Neural Networks (TWEANNs) is the fast learning Cartesian genetic programming evolved artificial neural networks (CGPANN) algorithm [11]. CGPANN has been demonstrated as a powerful approach in appraisals on the Wisconsin breast cancer (WBC) [22] data set as well as on the single and double pole balancing problems [1,12,13], warranting further evaluation of its use as a tool for bioinformatics. In this paper, we have identified a number of possible enhancements to the CGPANN approach which we will outline and evaluate on the WBC problem.

Section two discusses the WBC problem, introduces Cartesian genetic programming (CGP), and describes how it can be applied to produce ANNs. A short case study is provided of the previous application of CGPANN to the WBC data set. The third section outlines our experimental set up, including how the CGPANN algorithm was modified to incorporate RBF kernels and crossover. The performance of our novel CGPANN approach is then evaluated from several perspectives and the results discussed.

2 Previous Work

This section provides a case study of a previous application of CGPANN to the WBC problem. The WBC data set, Cartesian genetic programming and the CGPANN algorithm are first overviewed as pertinent background information to the study.

2.1 Wisconsin Breast Cancer Diagnosis Data Set

Excluding skin cancers, breast cancer is the most prevalent cancer among women [4]. The WBC data set consists of features generated from fine needle aspirate (FNA) biopsies of breast tumors using image processing [22]. Digital images are captured of stained FNA biopsies slides. Cell boundaries are isolated in each image and used to generate 30 attributes describing the mean, standard error and largest value for each of 10 nuclei features over the isolated cells; radius, perimeter, area, compactness, smoothness, concavity, concave points, symmetry, fractal dimension and texture. The process and selected features are further explained in the paper "*Nuclear feature extraction for breast tumor diagnosis*" [20]. The data set comprises 357 benign and 212 malignant expert classified exemplars.

2.2 Cartesian Genetic Programming

CGP is an evolutionary programming (EP) technique which represents computer programs as directed graphs [15]. The phenotype is organized into three sections; a layer of input nodes, a two-dimensional grid of processing nodes and a layer of output nodes. Each node has a unique identifier in numbered succession from the input to the output layer. A processing node, as depicted in Fig. 1, accepts

a number of connections from the input nodes and processing nodes in earlier columns specified by their unique identifiers, and maps these inputs to an output value using a function selected from a predefined list. Output nodes simply pass on the output of a single processing node. A simple CGP phenotype is demonstrated in Fig. 2.

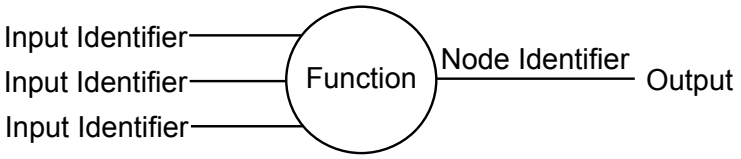


Fig. 1. Structure of a processing node in CGP. Input nodes and output nodes each implement a subset of the functionality of a processing node.

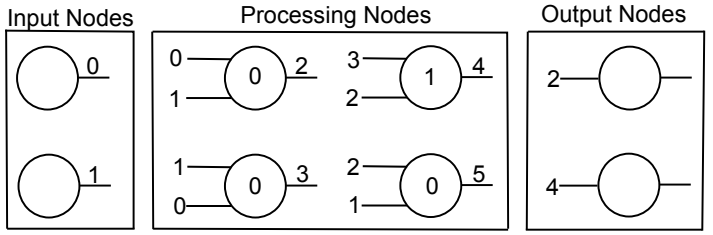


Fig. 2. A CGP system. Reading left to right, the nodes feed their outputs to nodes in later layers with corresponding input identifiers

A CGP genotype is a static length string of whole numbers. Each processing node is encoded using a single gene specifying the function it employs and a gene for the identity of each input. Output nodes are encoded as a gene corresponding to the identifier of the single processing node to which it connects. New CGP configurations can be created by mutating the function and input identifiers. Not all the processing nodes may be required to generate the output of the system (genes may be "*non-coding*").

CGP employs a "$1 + \lambda$" evolutionary strategy (ES). For a population of size n, λ is set to $n - 1$. Each generation, the single best performing solution is selected and λ children are generated from it using the mutation operator. The following generation comprises the best solution of the previous generation and its λ children.

2.3 Cartesian Genetic Programming Evolved Artificial Neural Networks

Khan *et al.* have adapted the CGP algorithm to a fast learning neuroevolution approach by mapping neural network parameters to the CGP genotype [13].

As already discussed, this approach has performed well on a number of benchmark problems. To allow CGP to function as a neural network, the following modifications are made:

- The mapping functions of the processing nodes are replaced with typical activation functions, such as the sigmoid, step, or tanh
- Each connection (the synapses) between the nodes (the neurons) receives an associated real numbered weight value

The output of a processing node is generated in a manner similar to a standard ANN neuron. The summation is taken of each input value (either the output of a processing node in an earlier column or an input node) multiplied by the associated weight. The sum is applied to the activation function to produce the output of the processing element. In the genotype, a processing node with k-inputs is represented by:

- A gene identifying which activation function to implement
- k genes encoding the input identifiers (as for CGP)
- k genes encoding the weights

2.4 Case Study: Breast Cancer Detection Using CGPANN

Ahmad *et al.* have recently reported good results for CGPANN on the WBC problem, outperforming a number of other published approaches[1]. The system accepts the 30 attributes as described for an exemplar to produce an output value. The output is rounded to 0 or 1 corresponding to a classification of benign or malignant respectively.

The approach uses a population of size 10, with a value of 9 for λ and a mutation rate of 10%. Evolution continues for 100,000 generations. Solution fitness is defined as the sum of true positives and true negatives on the training data. The parameters of the processing node grid were selected by trial and error over varying configurations with the number of columns ranging from 100 to 300 organised into a single row, with arity between 5 and 40.

The best performing configuration identified, which consisted of 200 columns with an arity of 35, was subjected to further evaluations. The architecture was able to achieve an average accuracy of 96% across 5 runs of 10 fold cross-validation on the entire data set. Although approaches exist with superior performance on this data set, CGPANN presented an error of only 1.5% on malignant exemplars.

3 Experimental Setup

Following from the encouraging work of Khan *et al.* and Ahmad *et al.*, we introduce a number of modifications to the CGPANN algorithm with the aim of improving efficiency and consistency. The power and quick learning ability of the CGPANN algorithm combined with the small number of exemplars means

that overfitting can occur rapidly [5]. To identify overfitting, select the best generalizing solutions, and provide a more accurate evaluation of the performance of the solutions generated in our experiments, we divide the available exemplars into three sets;

- **A training set** used to guide evolution
- **A testing set** for selecting the best generalizing solution
- **A validation set** on which the true generalization of solutions selected using the testing set can be evaluated

A separate test and validation set are required assuming that solutions can overfit the testing data given the limited number of exemplars and the large number of solutions evaluated.

A straight forward method of improving consistency is to average output across several runs. We implement this idea as a "*committee of machines*". Each committee member is trained independently using a different distribution of the available exemplars over the training and testing data sets. If the available data is randomly divided between the training and testing sets for each member of a committee of three machines, the training and testing data will typically each share roughly only one eighth of the exemplars across all three committee members, providing different views of the same data and reducing the impact of unfortunate sampling. Our experiments employed a committee of five machines.

3.1 Crossover

Evaluating CGPANN, it is apparent that the $1 + \lambda$ ES makes this a very greedy approach. Basing evolution on only the single best solution each generation means that it would be difficult to escape local minima unless promising evolutionary paths instantly display superior performance to their peers. Although such a greedy approach can provide quick learning on problems with small solution search spaces, where only a single minimum of the surface exists or where local minima are adequate, it is unlikely to scale well to problems requiring more complex solutions. Our approach to increasing the robustness of the ES is to incorporate crossover between the two best performing solutions each generation, which should provide a more intelligent search of the solution space.

The advantage and difficulty of crossover for the CGP algorithm has previously been demonstrated [3]. However, no such approach has been described or evaluated for CGPANN. We base our novel approach on that outlined by Clegg, Walker and Miller in the paper: "*A new crossover technique for Cartesian genetic programming*" for CGP networks with necessarily a single row [3]. The approach of Clegg *et al.* replaces the standard whole numbered genes used to specify node input IDs and function IDs with real numbers in the range $0 <= x < 1$. Using EQ 1 and EQ 2, gene values can be translated to whole numbered input and function identifiers respectively, equivalent to the standard CGP approach:

$$floor(gene_i \times nodeterm_j) \tag{1}$$

$$floor(gene_i \times func_{total}) \tag{2}$$

Where $gene_i$ is the value of the gene, $nodeterm_j$ is the unique identifier of the node, and $func_{total}$ is the number of available functions. As this approach is applied to CGPs with a single row, EQ 1 will translate the gene value to the identifier of a processing node in an earlier column up to but not including itself, or an input node. EQ 2 produces a number in the range of available functions.

Crossover is carried out by first generating a random "*contribution*", where $0 < contribution < 1$. A new child C can then be generated by applying EQ 3 to each gene g in two parent genomes P_1 and P_2:

$$C(g_i) = P_1(g_i) \times contribution + P_2(g_i) \times (1 - contribution) \tag{3}$$

Where $C(g_i)$ is the value of a gene in the offspring, and $P_1(g_i)$ and $P_2(g_i)$ are the values of the corresponding gene in each parent.

This approach can be extended to the CGPANN algorithm. As the synaptic weight genes are already encoded as real numbers, no mapping to a crossover compatible format is required. Crossover for synaptic weight genes can therefore be handled using just EQ. 3. The impact of the competing conventions problem [7] should be limited for CGPANN, as the greediness of the ES restricts how divergent any two selected parents can be. In our evaluations, we present the results for our approach both without crossover and with crossover at 40%.

3.2 Radial Basis Functions

From Table 1, it can be observed that approaches employing radial basis functions (RBFs) and the *k-neighbor* approach have enjoyed increased levels of success on the WBC data set [1] [8] [16]. It is also noted that the CGPANN approach may be favourable to RBF processing elements, as the topology evolving architecture allows for dynamic selection of the number of RBF centres and the consideration of different attributes by each, introducing an element of data mining to the approach. Following from this observation, we have adapted the CGPANN to act as an RBF network [18].

The genes representing the functions of the processing nodes are repurposed to specify the standard deviation for Gaussian functions. Therefore, instead of specifying different functions, the genes essentially specify different Gaussian function mappings. Euclidean distance is used to measure the distance between the input vector to a processing node and the centre of the RBF as specified by the synaptic weights. An additional real valued gene is added for each input which acts to scale the distance in each dimension, which should allow for more complex decision boundaries for the RBFs. The output neurons are modified to accept several weighted connections from the processing nodes, and implement a simple linear step or sigmoid function.

3.3 Algorithm

Contrary to the approach of Ahmad *et al.*, the error of the real valued output of the solutions is used to define fitness, as this is considered to provide a more

fine grained evaluation of performance. Each CGPANN solution comprises ten processing nodes in a single row, with an arity of 5. The use of fewer processing elements reduces the space of solutions which needs to be searched, producing more consistent results which can be optimised quickly. An independent population of size 10 is used to generate each committee member. The mutation rate and max perturbation of real valued genes are randomly set for each offspring in the range 0 to 0.2. Committee members are trained in turn until 500 generations pass where no performance increase is identified on the testing data set. In our experiments, this corresponds to an average of 2937 generations per CGPANN without crossover, or 3124 generations with crossover. The pseudocode for the algorithm is as follows:

```
While committee not filled
     Shuffle training and testing data
     While ‘‘generations no improvement’’ is less than 500
          Evaluate solutions on training data
          Evaluate generalization performance of best solution
          If generalization improved
               Reset ‘‘generations no improvement’’
               Record solution
          Else
               Increment ‘‘generations no improvement’’
          Populate next generation
     Save best generalising solution to committee
Evaluate committee on validation data
```

4 Results

The result of leave one out cross validation carried out across all 569 exemplars in the WBC data set for our modified CGPANN algorithm is given in Table 1 together with previously published results for a range of approaches on the same problem. Misclassifications are divided into two categories:

- **Type-1**: A benign case incorrectly classified as malignant
- **Type-2**: A malignant case incorrectly classified as benign

When calculating the type-1 and type-2 errors we consider only the number of corresponding misclassifications over the entire data set, e.g. type-1 errors are considered correctly classified when calculating the type-2 error. A breakdown of the performance of standard CGPANN and CGPANN using RBF processing elements is given in Table 2.

Table 1. Performance of various approaches on the WBC problem. Rows 1 to 20 taken from the paper "*Breast cancer detection using Cartesian genetic programming evolved artificial neural networks*" [1]. Rows 21 to 23 were generated in our experiments.

No.	Method	CCR
1	MLP	95.56 [6]
2	SVM	93.95
3	FLDA/MLP	90.92
4	PCA/MLP	92.02
5	GP/MDC	96.58
6	SOM-RBF	98.00 [17]
7	GR	96.76
8	RBF	97.04
9	MOE	96.29
10	LDA	96.34
11	Logistic	97.22
12	K neighbour	96.78
13	Kernel	95.02
14	GP test average	96.32
15	L2-SVM/GDVEE(RBF)	**98.10 [8]**
16	SVM (Linear)	94.00
17	SVM (RBF)	97.70
18	Fuzzy	95.80
19	ENN	95.60
20	CGPANN	96.0
21	MFF-NEAT [14]	96.49
22	CGPANN-RBF	97.19
23	CGPANN-RBF with Crossover	97.19

The performance of three different approaches to combining the output of the committee to present a single unified response are presented in Table 3. The three approaches are:

- **Average**: the unweighted average of the outputs of all committee members
- **Voting**: majority voting
- **Confidence**: selecting the output of the committee member with the highest evidential response [9]

5 Discussion of Results

Averaging the outputs of the committee of machines, the performance of the modified CGPANN algorithm is positive and encouraging, outperforming a number of other published approaches as referenced in Table 1, and ameliorating CGPANN performance on the problem while simultaneously improving efficiency. The results demonstrate that through our modifications, CGPANN is capable of successfully producing RBF networks.

Table 2. Performance of standard CGPANN and our modified approach in terms of correctly classified records (CCR) and a breakdown of performance on type-1 and type-2 error

	CGPANN	CGPANN-RBF	CGPANN-RBF with Crossover
CCR	96	97.19	97.19
Type-1 Error	0.0300	0.0088	0.0088
Type-2 Error	0.0150	0.0193	0.0193

Table 3. The correct classification rate for each of three approaches for combining the output of the committee members

	With Crossover	Without Crossover
Average	97.2	97.2
Confidence	97.0	96.8
Voting	97.0	97.2

Discounting the advantages of the committee architecture, the committee members considered individually could correctly classify 96.8%, 96.5%, 96.3%, 95.6% and 95.6% of the entire dataset when using crossover at 40%, or 96.8%, 96.5%, 96.3%, 96.3% and 96.3% without crossover. Although this is quite a small sampling, eight of the ten networks were able to outperform the standard CGPANN model, suggesting that the use of RBF processing nodes does itself offer an advantage in solving this problem. The results also highlight the advantage of using a committee of machines where the probability of achieving good performance is higher than that of underperforming. However, further tests are needed to prove that the use of a committee of machines is statistically significant, and to identify the minimum committee size required to provide consistently good performance with the desired level of confidence.

The efficiency of CGPANN also appeared to benefit from our modifications, as each committee member was trained for an average of only 3031 generations. As efficiency in training in this situation is less important than the overall results achieved, it may be beneficial to sacrifice efficiency in favor of increased search space coverage, and hopefully performance increases. For example, instead of stopping training when no improvement is identified over a long period, it may be more beneficial to increase mutation rates and decrease crossover rates to improve the chance of escaping local minima.

The impact on performance afforded by the use of crossover in our experiments is less encouraging; the average CCR was actually observed to decrease from 96.44% to 96.16% on individual CGPANN systems when employing crossover at 40%. This data suggests a decrease in consistency and performance when using crossover, but applying a two sample T-test to the results produces a t-value of 1.0722, which is insufficient to assert that crossover (as implemented here) decreases performance. The distribution of solution fitness values in our experiments for 5 CGPANNs created using crossover and 5 CGPANNs without crossover is provided in box chart format in Fig. 3.

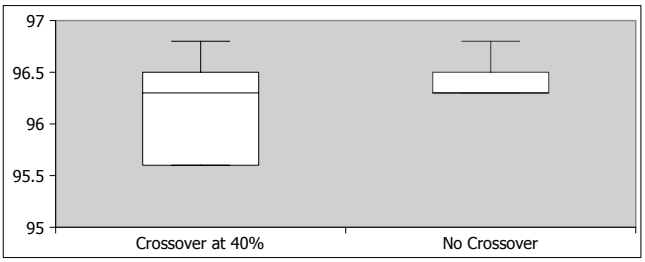


Fig. 3. Distribution of fitness values of five CGPANNs trained with and without crossover

A type-2 error may lead to a missed diagnosis and be detrimental to patient prognosis. A type-1 error calls for further patient tests to be carried out, and as such is considered a low-cost error type. Our experiments were aimed at minimizing overall error, whereas in the previously published work, Ahmad *et al.* selected parameters through trial and error specifically geared towards reducing type-2 error. Although our modified CGPANN approach showed superior performance, 11 of the 16 misclassified exemplars were malignant. This represents an increase of almost 30% type 2 error (irrespective of whether crossover is employed or not) relative to the results published by Ahmad *et alii.*

A contributing factor to the high type-2 error may be the relative abundance of benign exemplars. Therefore, focus on correctly classifying the benign exemplars is more rewarding for evolutionary algorithms (in terms of fitness), a problem which would be exacerbated by the greediness of the CGPANN algorithm. This problem may be redressed, perhaps, through providing fitness bonuses for the correct classification of malignant exemplars, or by incorporating algorithms such as boosting and bagging into the process [19].

In our experiments, all five committee members generated using crossover incorrectly classified 8 exemplars. One of these exemplars was randomly selected and subjected to further informal evaluations using the CGPANN networks, various commercially available neural network packages and the MFF-NEAT algorithm, with varying parameters and distributions of the training and testing data. All trialled approaches failed to correctly classify this exemplar. This does not necessarily suggest that this exemplar is incorrectly labelled, but perhaps the reasoning behind its classification if not adequately described by the data or represented in the data set. Additionally, the presence of this exemplar (and perhaps others) in the training data could inhibit successful learning of the remaining exemplars. Therefore, it is considered that further increases in performance may be best broached through evaluation and validation of the data, rather than through further development of machine learning approaches.

6 Conclusions and Future Work

In this paper we have demonstrated that the performance of the CGPANN on the Wisconsin breast cancer problem can be improved through the use of RBF processing elements. Although consistency is still an issue, our evaluations demonstrate that it can be somewhat abated through the use of a committee of machines. Our results also suggest that, while not definitive, crossover in the form described here can be detrimental to the performance and efficiency of CGPANNs implemented using RBFs. The SVM can however still be regarded as the state of the art for this problem.

Future work in this area, in addition to issues and ideas raised in the discussion section, will evaluate an avenue of investigation showing positive potential, but which is not fully investigated at this time. This idea involves starting with an extremely large committee size, and purging committee members with low generalization ability at regular generation intervals. This approach leverages the quick learning ability of CGPANN to quickly discount unpromising evolutionary avenues to provide a more expansive and intelligent search of the solution space, and hopefully lead to further improvements in consistency and performance.

Acknowledgments. This work was supported by a Cork Institute of Technology Rísam scholarship. Dr Paul Walsh, CTO NSilico is a Principal Investigator on ClouDx-i an FP7-PEOPLE-2012-IAPP.

References

1. Ahmad, A.M., Khan, G.M., Mahmud, S.A., Miller, J.F.: Breast cancer detection using cartesian genetic programming evolved artificial neural networks. In: Soule, T. (ed.) Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference (GECCO 2012), pp. 1031–1038. ACM, New York (2012)
2. Bari, A., Bhasin, K., Karnawat, D.N.: Introduction to Neural Network and Improved Algorithm to Avoid Local Minima and Faster Convergence. In: Das, V.V., Thankachan, N. (eds.) CIIT 2011. CCIS, vol. 250, pp. 396–400. Springer, Heidelberg (2011)
3. Clegg, J., Walker, J.A., Miller, J.F.: A new crossover technique for cartesian genetic programming. In: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, pp. 1580–1587. ACM, New York (2007)
4. DeSantis, C., Siegel, R., Bandi, P., Jemal, A.: Breast cancer statistics. CA-Cancer J. Clin. 61, 408–418 (2011)
5. Domingos, P.: A few useful things to know about machine learning. Commun. ACM 55, 78–87 (2012)
6. Guo, H., Nandi, A.K.: Breast cancer diagnosis using genetic programming generated feature. Pattern Recogn. 39(5), 980–987 (2006)
7. Hancock, P.J.: Genetic algorithms and permutation problems: A comparison of recombination operators for neural net structure specification. In: International Workshop on Combinations of Genetic Algorithms and Neural Networks, COGANN 1992, pp. 108–122. IEEE (1992)

8. Iranpour, M., Almassi, S., Analoui, S.: Breast cancer detection from fna using svm and rbf classifier. In: 1st Joint Congress on Fuzzy and Intelligent Systems (2007)
9. Jain, B.J., Wysotzki, F.: Efficient Pattern Discrimination with Inhibitory WTA Nets. In: Dorffner, G., Bischof, H., Hornik, K. (eds.) ICANN 2001. LNCS, vol. 2130, pp. 827–834. Springer, Heidelberg (2001)
10. Jung, J.-Y., Reggia, J.A.: The Automated Design of Artificial Neural Networks Using Evolutionary Computation. In: Yang, A., Shan, Y., Bui, L.T. (eds.) Success in Evolutionary Computation. SCI, vol. 92, pp. 19–41. Springer, Heidelberg (2008)
11. Khan, M., Khan, G.: A novel neuroevolutionary algorithm: Cartesian genetic programming evolved artificial neural network (cgpann). In: Proceedings of the 8th International Conference on Frontiers of Information Technology (FIT 2010), pp. 48:1–48:4. ACM, New York (2010)
12. Khan, M.M., Khan, G.M., Miller, J.F.: Efficient representation of recurrent neural networks for markovian/non-markovian non-linear control problems. In: 2010 10th International Conference on Intelligent Systems Design and Applications (ISDA), pp. 615–620. IEEE (2010)
13. Khan, M., Khan, G., Miller, J.: Evolution of neural networks using Cartesian genetic programming. In: 2010 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8. IEEE (2010)
14. Manning, T., Walsh, P.: Automatic Task Decomposition for the NeuroEvolution of Augmenting Topologies (NEAT) Algorithm. In: Giacobini, M., Vanneschi, L., Bush, W.S. (eds.) EvoBIO 2012. LNCS, vol. 7246, pp. 1–12. Springer, Heidelberg (2012)
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. Moriarty, D.E.: Symbiotic evolution of neural networks in sequential decision tasks. PhD thesis, University of Texas at Austin (1997)
17. Mu, T., Nandi, A.K.: Breast cancer detection from fna using svm with different parameter tuning systems and som–rbf classifier. Journal of the Franklin Institute-engineering and Applied Mathematics 344(3), 285–311 (2007)
18. Park, J., Sandberg, I.W.: Universal approximation using radial-basis-function networks. Neural Comput. 3(2), 246–257 (1991)
19. Quinlan, R.: Bagging, boosting, and c4.5. In: Proceedings of the National Conference on Artificial Intelligence, pp. 725–730. AAAI Press (1996)
20. Street, N., Wolberg, W.H., Mangasarian, O.L.: Nuclear feature extraction for breast tumor diagnosis. University of Wisconsin-Madison, Computer Sciences Department (1992)
21. Tetko, I.V., Livingstone, D.J., Luik, A.I.: Neural network studies. 1. Comparison of overfitting and overtraining. Journal of Chemical Information and Computer Sciences 35(5), 826–833 (1995)
22. Wolberg, W.H., Mangasarian, O.L.: Multisurface method of pattern separation for medical diagnosis applied to breast cytology. Proceedings of the National Academy of Sciences 87(23), 9193–9196 (1990)

An Evolutionary Approach to Wetlands Design

Marco Gaudesi[1], Andrea Marion[2], Tommaso Musner[2],
Giovanni Squillero[1], and Alberto Tonda[3]

[1] Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy
{marco.gaudesi,giovanni.squillero}@polito.it
[2] University of Padova, Via Marzolo 9, 35131 Padova, Italy
andrea.marion@unipd.it, tommaso.musner@studenti.unipd.it
[3] UMR 782 GMPA, INRA, 1 Av. Brétignières, 78850 Thiverval-Grignon, France
alberto.tonda@grignon.inra.fr

Abstract. Wetlands are artificial basins that exploit the capabilities of some species of plants to purify water from pollutants. The design process is currently long and laborious: such vegetated areas are inserted within the basin by trial and error, since there is no automatic system able to maximize the efficiency in terms of filtering. Only at the end of several attempts, experts are able to determine which is the most convenient configuration and choose up a layout. This paper proposes the use of an evolutionary algorithm to automate both the placement and the sizing of vegetated areas within a basin. The process begins from a random population of solutions and, evaluating their efficiency with an state-of-the-art fluid-dynamics simulation framework, the evolutionary algorithm is able to automatically find optimized solution whose performance are comparable with those achieved by human experts.

Keywords: Evolutionary Algorithms, Wetlands Design, Ecological Modelling.

1 Introduction

Nowadays, more and more specialists are becoming involved in pollution control, one of the biggest problem of our time. Ecosystems are stressed by pollution; organic chemicals, while contributing to their destruction, can also make the water not usable by animals and humans. To bring down the quantity of chemicals dissolved in water, researchers proposed a new approach, based on bio-geochemical processes naturally present in the environment, adopting *free surface constructed wetlands*. A wetland consist of a small artificial basin, partially flooded with water and containing many vegetated areas, in which the water flows and undergoes a natural filtering process from pollutants due to particular plant species, which are able to use these waste products to support their vital functions (e.g., *Phragmites Australis, Typha Latifolia*); vegetated areas have to be distributed over the wetland in order to increase the filtering performance. In the last half century a great effort in wastewater treatment has been performed with special plants able to process polluted water. It has been demonstrated that this approach is more

L. Vanneschi, W.S. Bush, and M. Giacobini (Eds.): EvoBIO 2013, LNCS 7833, pp. 177–187, 2013.

useful with *point sources*, characterized by little quantities of fluid polluted by high concentrations of chemicals, rather than *diffused sources*, characterized by big quantities of fluid polluted by low concentrations of chemicals.

To design a wetland, experts create several configurations which are then processed by a tool to simulate the flow of water and to calculate the efficiency in terms of filtering by a hydrodynamic simulator, used to numerically solve the velocity field and to calculate the wetland removal efficiency produced by a specific wetland configuration in terms of bathymetry, vegetation distribution, hydraulic pathways. The classic *trial and error* approach is the only viable one, since it is not possible to implement an inverse function able to identify with precision positions and characteristics of each vegetated area to be inserted in the basin, in order to obtain an optimum filtering capability.

The proposed idea is to evolve a population of individuals, each one representing a set of vegetated patches distributed over the two-dimensional wetland domain. The evolutionary approach is autonomously able to optimize the performance of the wetland, while an appropriate set of constraints enforces realistic configurations. The preliminary study of a system able to automatically calculate solutions for the problem was verified in [9]. Here, the goal it to tackle a realistic problem by include different constraints.

2 Wetlands

Cowardin [5] defines a wetland as an ecosystem transitional between aquatic and terrestrial ecosystems, in which the water table is usually at or near the surface or the land is covered by shallow water [4]. Before the extensive land reclamation through the last century, wetlands were common along the coasts, where they functioned as a natural buffer between inner agricultural zones and coastal areas. Today there is a pressing necessity to restore these areas and their role, defining optimal design criteria to obtain, at reasonable costs, the best removal efficiency.

The removal efficiency of natural and constructed free-surface wetlands is controlled by the time spent by contaminants into vegetated zones [15]. The role of vegetation in wetlands is important for two main reasons: water passing through vegetated zones decreases its local velocity, favoring the sedimentation of suspended solids; and biochemical processes determine a transformation of the dissolved substances. In combination with bathymetry, distribution of vegetation can produce preferential pathways of water (hydraulic shortcuts) that can substantially decrease the overall efficiency of a wetland. Removal efficiency is also affected by other hydrodynamic characteristics, as water depth and discharge, both dependent on vegetation distribution and density [1] [12]. Wetlands constructed for waste water treatment are often designed considering an average water residence time [12], even though these methods cannot adequately describe spatial configurations of vegetation in real wetlands [13]. These models, usually called *zero-dimensional*, are often used because they require few data and are easy to manage. Nevertheless, zero-dimensional models produce significant inaccuracies in the prediction of the efficiency of contaminant removal. Other *one-dimensional* models with transient

storage were recently used [14] to assess the contaminant removal in a constructed wetland, giving in most cases a good approximation of breakthrough curves.

These models, however, fail to describe different flow paths across the vegetation and through main channels. The evidence of different flow pathways results in a clear bimodality of the solute breakthrough curves, that account for the different characteristic time scales of water residence time. Since spatial heterogeneity of the variables assumes a prominent role in determining the removal efficiency, the use of a more detailed *two-dimensional* approach becomes necessary to obtain reliable predictions.

3 Proposed Approach

In the proposed approach the design of a wetland is fully automated exploiting an evolutionary algorithm. Each individual of the population represents a complete configuration of the wetland, expressed as a set of patches of vegetation arranged within the area of the basin; each vegetated area is defined by its position and diameter. The evolutionary algorithm handles the creation and evolution of individuals, while the actual evaluation is performed by a tool able to simulate the flow of water within the wetland and calculate the filtering capacity. Differently from the feasibility study, candidate solutions has been provided more stringent constraints in order to evolve towards optimized solutions close to a real ones. This constraint has been applied to the maximum area that can be covered by vegetation patches; the limit was set to 60%, in order to push the evolution towards the realization of optimized individuals describing more closely a configuration similar to those that are actually made.

3.1 Mathematical Models

A wetland is modeled using a two-dimensional depth averaged model that solves hydrodynamics, coupled with a two-dimensional solute transport equation with a first order decay term. Under the assumption of hydrostatic pressure, stationary flow, and negligible wind and Coriolis forces, the depth-averaged velocity field and water depth can be described by the following equations [19]:

$$\frac{\partial(hU)}{\partial x} + \frac{\partial(hV)}{\partial y} = 0 \tag{1}$$

$$\frac{\partial(hU^2)}{\partial x} + \frac{\partial(hUV)}{\partial y} = = -gh\frac{\partial z_s}{\partial x} + \frac{1}{\rho}\frac{\partial(hT_{xx})}{\partial x} + \frac{1}{\rho}\frac{\partial(hT_{xy})}{\partial y} - \frac{\tau_{bx}}{\rho} \tag{2}$$

$$\frac{\partial(hUV)}{\partial x} + \frac{\partial(hV^2)}{\partial y} = = -gh\frac{\partial z_s}{\partial y} + \frac{1}{\rho}\frac{\partial(hT_{yx})}{\partial x} + \frac{1}{\rho}\frac{\partial(hT_{yy})}{\partial y} - \frac{\tau_{by}}{\rho} \tag{3}$$

The quantities U and V represent the depth-averaged velocities $[m\,s^{-1}]$ along the x and y direction, respectively, h is the water depth $[m]$, z_s is the water

surface elevation $[m]$, and ρ the water density $[kg\,m^{-3}]$. The bed shear stresses τ_{bx} and τ_{by} $[N\,m^{-2}]$ in the x and y direction respectively are calculated using the following relationships:

$$\tau_{bx} = \rho c_f m_b U \sqrt{U^2 + V^2} \tag{4}$$

$$\tau_{by} = \rho c_f m_b V \sqrt{U^2 + V^2} \tag{5}$$

In the case modeled here, the bed slope is set to zero and the investigated velocity range makes it possible to consider the friction coefficient as a constant. This assumption generally holds where the velocity is sufficiently fast to assume turbulent flow. For a flat bathymetry, the bed slope coefficient m_b is unitary and the coefficient of friction c_f can be rewritten using Manning equation as $c_f = gn^2 h^{-1/3}$. The effect of different vegetation densities is modeled here using different values of Manning roughness coefficient. This choice is confirmed by many studies that relate vegetation density, stem diameter and flow conditions to an equivalent roughness coefficient [3] [10] [17]. Fluid shear stresses $T_{ij}(i, j = x, y)$ associated to viscous and turbulent effects, are determined using the Boussinesq assumption:

$$T_{xx} = 2\rho(\nu + \nu_t)\frac{\partial U}{\partial x} \tag{6}$$

$$T_{xy} = T_{yx} = \rho(\nu + \nu_t)(\frac{\partial U}{\partial y} + \frac{\partial V}{\partial x}) \tag{7}$$

$$T_{yy} = 2\rho(\nu + \nu_t)\frac{\partial V}{\partial x} \tag{8}$$

where ν, ν_t, are the kinematic and eddy viscosities $[m^2\,s^{-1}]$. Since the kinematic viscosity has a lower value than the eddy viscosity, it can be neglected in most cases. For a turbulent flow regime, as it was assumed in this preliminary study, ν_t can be expressed using Elder depth-averaged parabolic model [6] as $\nu_t = \alpha U_* h$, where the term α is an empirical coefficient $[-]$ and U_* is the shear velocity $[m\,s^{-1}]$. For longitudinal dispersion Elder proposed a value of the coefficient α of about 5.9 [6], for transverse dispersion, Fischer found that α varies between 0.3-1.0 in irregular waterways with weak meanders [7]. In accordance with [2] [19] a value of 6.0 and 0.6 was chosen for the longitudinal and transversal dispersion coefficients respectively.

Solute transport of a reactive tracer through the wetland is simulated with a depth-averaged solute transport model accounting for the effect of advection, turbulent diffusion, dispersion and decay. In the simulations, the tracer is assumed to interact with vegetation and the chemical breakdown due to the permanence in the vegetated zones is modeled with a first order decay relationship. The equation governing the transport of a reactive tracer in the wetland can be modeled as:

$$\frac{\partial(hUC)}{\partial x} + \frac{\partial(hVC)}{\partial y} = = \frac{\partial}{\partial x}(hE_x\frac{\partial C}{\partial x}) + \frac{\partial}{\partial y}(hE_y\frac{\partial C}{\partial y}) - h\lambda C \tag{9}$$

where C is the depth-averaged solute concentration $[kg\,m^{-3}]$, U, V are the vertically integrated velocity components under steady flow conditions $[m\,s^{-1}]$ in the x, y directions respectively. Coefficient E_x, E_x $[m^2\,s^{-1}]$, account for both turbulent diffusion and dispersion. A constant homogeneous value of E_x, E_y is chosen $(10^{-5}\,m^2\,s^{-1})$ throughout the entire domain.

3.2 Evolutionary Core

The EA used is μGP [16], it is a versatile toolkit developed at Politecnico di Torino in the early 2000s and available under the GNU Public License from Sourceforge[1]. μGP original use was to assist microprocessors' designers in the generation of programs for testing and verification, hence, the Greek letter mu in its name. But over the years has been used as optimizer in a much wider spectrum of problems, including numerical optimizations.

The algorithm initially creates a set of random candidate solutions to the given problem, that are then evaluated, and sorted by their fitness value (see Subsection 3.3). Offspring is then created favoring the fittest individuals and also trying to favor diversity among the population. New candidate solutions are then evaluated and added to the initial population. Solutions are again sorted, and the worst ones are removed until the population returns to its original size. The process is then iterated, starting from offspring generation, until a stop condition is reached.

Two categories of genetic operators are used to generate the offspring: *mutations*, or single-parent operators, and *crossovers*, or recombination operators. Mutation operators create new candidate solutions by altering one single parent solution; crossover operators mix the information contained in two or more parents solutions to create offspring. The most common operators are available inside μGP, but the toolkit also implements *differential evolution*, and other operators specially calibrated for real parameters.

Individuals are internally represented as a multigraph, μGP relies on a external configuration file constraints the multigraphs to sensible structure, and maps the internal individuals to valid solutions of the problem. In the specific context, each individual encodes a candidate configurations of the wetland, that is, it details the features of the several patches of vegetation, with variable number of occurrences from 20 to 35, that are going to be placed in the water; the order in which the patches are described within the individual is irrelevant. All patches are assumed to be of circular shape. Since they can overlap, however, they can create more complex shapes. A patch is characterized by its position (x, y coordinates expressed in real values) in the wetland and its radius; in this simplified approach friction value is always the same. A patch's position is constrained by the size of the wetland; its radius is constrained following the minimum and maximum size of actual patches of vegetation used in real wetlands.

[1] `http://ugp3.sourceforge.net/`

3.3 Fitness Function

The definition of an appropriate fitness function is a key aspect in the use of an EA. The process of evolution is based on *differential survival*, that is, different individuals must have a different chance to spread their offspring in future generations related to their fitness. Thus in the artificial environment modeled by an EA, it is essential that different individual get different fitness values. It is a common practice to include in the fitness some heuristic knowledge, in order to help the EA to explore the most promising regions of the search space.

In μGP, the fitness is not a single value but a vector of positive coefficients. The individual A is considered to be fitter than the individual B if the first j elements of the two fitness vectors are equals, and the $(j+1)-th$ element of the A's fitness is greater than the $(j+1)-th$ element of the B's fitness. In the context of wetland optimization, three values have been used.

In order to evaluate the goodness of a candidate wetland layout, a simulation of the hydrodynamic field is performed extracting computed values of discharge $Q\,[m^3\,s^{-1}]$ and water depth h at the inlet and at the outlet sections of the wetland. During the simulation, a *reactive tracer* with a known concentration is injected at the inlet. Thanks to the presence of vegetation the tracer is gradually degraded and reaches the outlet section. Mass flux $\hat{M}\,[kg\,s^{-1}]$ passing through these sections is measured, and the difference between the two values represent the first parameter of the fitness function. In order to obtain the optimal vegetation distribution, this difference must be maximized.

On the other hand, a candidate layout must still let the water flow, avoiding configurations where the vegetation is so dense to make the flow impossible. The energy requested by the water to flow can be represented by the difference between the water depth at the inlet and outlet section. This difference represents the second parameter of the fitness function. This parameter is minimized by the algorithm: solutions that completely block the water flow are then heavily penalized.

The third and last fitness parameter measures the difference of discharge between the inlet and the outlet sections of the wetland. This value assures that the stationary flow conditions are reached and that the mass fluxes are finely computed. This discharge difference is strongly minimized.

4 Experimental Evaluation

4.1 Setup

The artificial basin take into consideration in this work has a rectangular shape with dimensions $200m$-long-by-$100m$-wide, with a water depth considered constant over the entire surface and equal to $0.5\,m$. The inlet and outlet sections are located at the centre of the shorter sides of the wetland and have 10 m of size amplitude. In this way can be reached two important objectives: the first, related to the proportions of the area, concerns the total spread of the incoming water flow over the entire section of the basin; the second, due to the constant

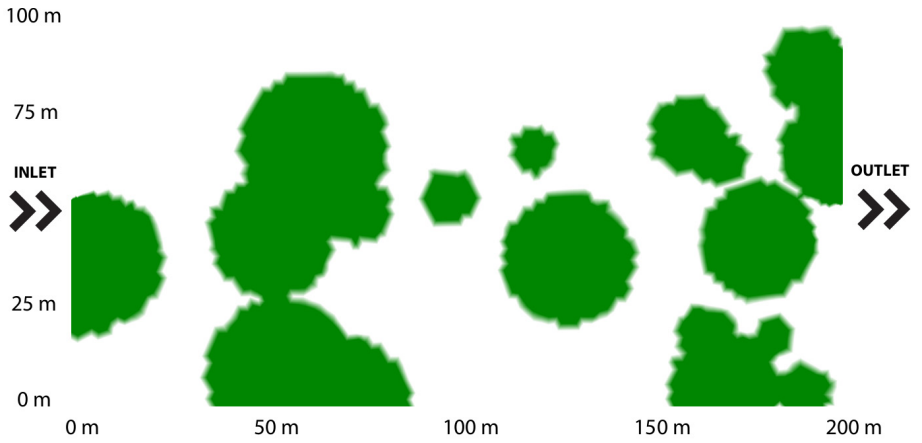


Fig. 1. Individual B: Representation of the phenotype of an individual extracted from the first generation of evolution; dark areas show the distribution of vegetation over the wetland surface

depth, makes this basin more similar to the natural ones and also makes it possible to simplify the system, which will not consider any slopes of the basin's bed [18] [19]. In addition, a constant discharge of $0.2 m^3 s^{-1}$ is imposed at the inlet section. The rest of the wetland was considered impermeable and laws of friction have not been applied at the side walls. In order to monitor the filtering process of the wetland, within the inlet section is injected a reactive solute with a constant concentration of $1 kg m^{-3}$; in this way it is possible to extract the fitness value (which indicates the filtering capability of the basin) by calculating the average value of the concentration of this reagent in outlet area.

In order to simulate the hydrodynamic flow within the basin and the correct values of decay related to pollutants, it has been necessary to set some parameters into the fluid dynamics simulator tool. The basin was defined through an adaptive triangular mesh, so as to ensure a sufficient numerical stability and the required resolution in case of steep gradients of the hydrodynamic and solute transport solutions. In addition, was applied to each node a value of the Manning roughness coefficient and a decay value, depending on the structure of each individual. For this experiment, a particular configuration with a maximum vegetation cover of 60 % (expressed as vegetated surface over the whole wetland surface) and a single density for each vegetation patch were imposed. In conclusion, it was chosen to apply a single law of decay to a node of the mesh in which there is a vegetation patch, or a zero coefficient otherwise; it was chosen a decay coefficient equal to $5^{-6} s^{-1}$. In the same way, Manning roughness coefficients are set to $0.20 s m^{-\frac{1}{3}}$ to nodes with vegetation, and $0.02 s m^{-\frac{1}{3}}$ otherwise.

As previously introduced, to achieve this automatic optimization system were used two different tools, both open-source and freely available on internet. The tool used for evolutionary algorithm is μGP version 3.2.0 (revision 198). To simulate and evaluate each individual instead was used a tool called *TELEMAC2D*,

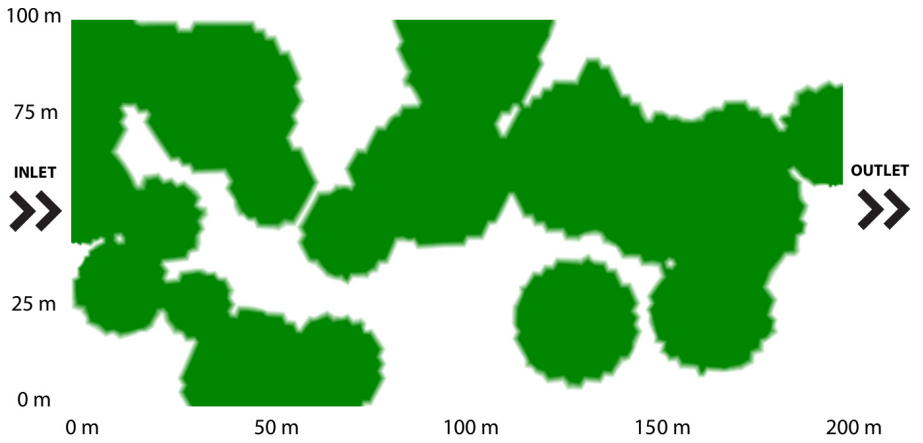


Fig. 2. Individual 7: Individual with percentage of vegetation next to the maximum limit but without good filtering performance, due to the distribution not optimized within the basin

part of the wider set of programs *openTELEMAC* [8] [11]. The code of the latter has been specifically modified in order to extract information relating to fitness in the format required by the μGP tool.

Each individual evolved by the evolutionary tool is converted to the *TELEMAC2D* format, that consists of a map of basin's nodes, and each of these nodes can be covered or not by a vegetation patch. For this reason, each individual undergoes a sort of pre-processing that inserts in the nodes of the map values associated with vegetated areas. The process has been elaborated on a single machine, equipped with an *Intel Core i7-950* CPU running at 3.06 GHz, and the whole system was setting in order to process up to 4 individuals simultaneously, with an average computation time of 90 minutes for each individual.

4.2 EA Configuration and Result Discussion

In order to obtain the results described in this paper, the EA has been configured in such a way to create a random initial population of 20 individuals ($\mu = 20$), on which they are applied, at each generation of the evolution, 11 genetic operators ($\lambda = 12$) chosen among the 20 available in μGP tool. Two different types of genetic operators are used: standard crossover operators with one or two cut points and nine mutation operators able to insert, remove or exchange small parts of genome between individuals. Further details concerning the genetic operators available in μGP tool can be found in [16]. The entire process evolved for 90 generations, for a total of 1070 individuals generated. During the conversion of individuals to the format compatible with *TELEMAC2D*, a certain percentage of them was discarded because it was violating the introduced constraint about maximum area that vegetation patches can cover.

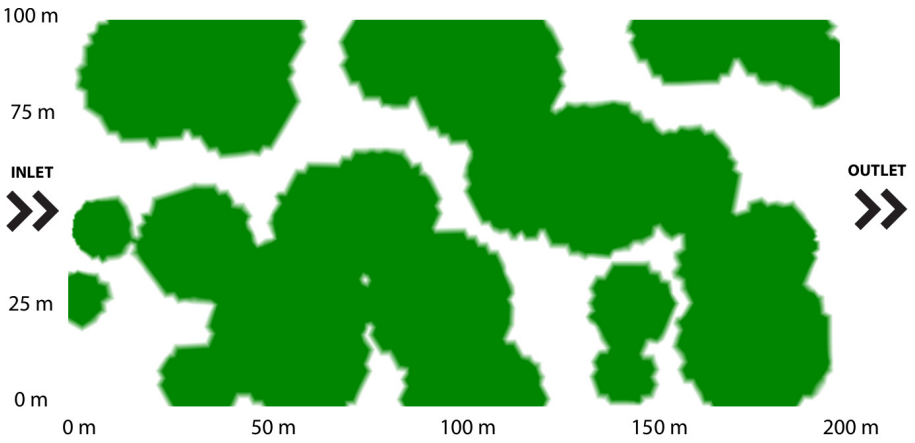


Fig. 3. Individual AAU: Representation of the individual that reached the best optimization level. The percentage of vegetation is close to the imposed limit to 60% but, thanks to the best arrangement of vegetation patches, its filtering performance is optimal.

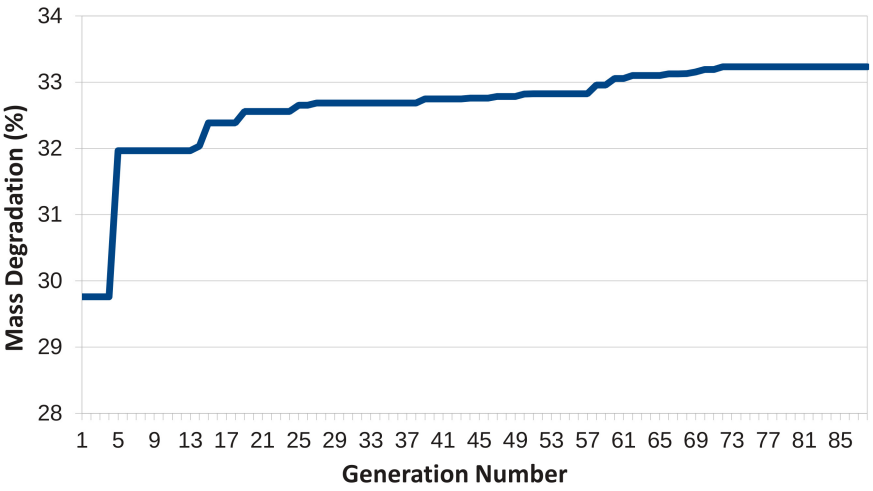


Fig. 4. A graph showing the trend of the best fitness value during generations

Starting from a random population, the evolution has shown several interesting features, which show the actual goodness of this approach. Among individuals of first generations, its possible to find some as the individual B which are formed by a low number of vegetated areas clearly separated between them, configuration that shown a low filtration capacity; in particular, the configuration shown in Figure 1 ensures a performance of pollutant reduction of 21% respect to the inlet concentration. As evolution proceeds, the trend of evolutionary

algorithm to generate individuals increases. This feature respect the constraint of the maximum coverage and, using the maximum available number of patches, the EA is able to combine them to create complex shapes able to modify the water flow and to optimize the filtering performance.

Figures Figure 2 and Figure 3 compare the two individuals 7 and *AUU*, both characterized by a vegetated coverage very close to the imposed limit of 60%, but with different fitness. Individual 7 belongs to the third generation, in which evolution is still very close to the starting stage and, despite the use of maximum coverage allowed, performances in terms of filtering amount to 27%. Individual *AUU* instead represents the best configuration achieved in this experiment, comparable to previous in terms of vegetated area; in this case the filtering capacity has been optimized to achieve performances of 33.2%. In the figure Figure 4 is shown a graph that describes the trend of fitness values during generations.

5 Conclusions

Wetlands are artificial ponds, and nowadays are extensively used to filtrate and purify water. Optimizing their design is an extremely complex task, and it is currently carried on by experts using a trial-and-error approach on the basis of fluid-dynamics simulations. In this paper, an evolutionary algorithm is applied to the wetlands design problem. Each candidate solution is evaluated by a state-of-the-art fluid-dynamics simulator, on the basis of several relevant metrics. Experimental results of the best solution provided by the algorithm show a performance comparable with human-devised designs, despite the absence of human intervention during the optimization process.

Future works will include a more complex individual representation, with patches of several different shapes and a more refined management of friction values. Managing larger populations, or different sub-population, might also prove beneficial to the quality of the final solutions: nevertheless, the computational-intensive simulations needed to evaluate a single candidate represent a severe bottleneck. For this reason, further developments will probably exploit the parallelism innate in evolutionary algorithms, using clusters or grids to speed up the process. Finally, the choice of decay coefficients has a predominant role in determination of the final breakdown efficiency: a more detailed analysis on a real case should be used to demonstrate the potential of the proposed approach, that shows promising results in this first experience.

References

1. Akratos, C.S., Tsihrintzis, V.A.: Effect of temperature, HRT, vegetation and porous media on removal efficiency of pilot-scale horizontal subsurface flow constructed wetlands. Ecological Engineering 29(2), 173–191 (2007)
2. Arega, F., Sanders, B.F.: Dispersion Model for Tidal Wetlands. Journal of Hydraulic Engineering 130(8), 739–754 (2004)
3. Augustijn, D.C.M., Huthoff, F., Velzen, E.H.: Comparison of vegetation roughness descriptions (2006)

4. Bendoricchio, G., Jorgensen, S.E. (eds.): Fundamentals of Ecological Modelling, 3rd edn. Elsevier Science (August 2001)
5. Cowardin, L.M.: Classification of Wetlands and Deepwater Habitats of the United States. DIANE Publishing (1979)
6. Elder, J.W.: The dispersion of marked fluid in turbulent shear flow. J. Fluid Mech. 5(4), 544–560 (1959)
7. Fischer, H.B.: Mixing in inland and coastal waters. Academic Pr. (1979)
8. Galland, J.C., Goutal, N., Hervouet, J.M.: TELEMAC: A new numerical model for solving shallow water equations. Advances in Water Resources AWREDI 14(3) (1991)
9. Gaudesi, M., Marion, A., Musner, T., Squillero, G., Tonda, A.: Evolutionary Optimization of Wetlands Design. In: 28th Symposium on Applied Computing (SAC) (2013)
10. Green, J.E.P., Garton, J.E.: Vegetation lined channel design procedures. Transactions of the American Society of Agricultural Engineers 26(2), 437–439 (1983)
11. Hervouet, J.M., Hubert, J.L., Janin, J.M., Lepeintre, F., Peltier, E.: The computation of free surface flows with TELEMAC: an example of evolution towards hydroinformatics. Journal of Hydraulic Research 32(S1), 45–64 (1994)
12. Kadlec, R.H., Wallace, S.: Treatment wetlands. CRC (2009)
13. Kadlec, R.H.: The inadequacy of first-order treatment wetland models. Ecological Engineering 15(1-2), 105–119 (2000)
14. Martinez, C.J., Wise, W.R.: Analysis of constructed treatment wetland hydraulics with the transient storage model OTIS. Ecological Engineering 20(3), 211–222 (2003)
15. Persson, J., Somes, N.L.G., Wong, T.H.F.: Hydraulics efficiency of constructed wetlands and ponds. Water Science & Technology 40(3), 291–300 (1999)
16. Sanchez, E., Schillaci, M., Squillero, G.: Evolutionary Optimization: the μGP toolkit, 1st edn. Springer (April 2011)
17. White, B.L., Nepf, H.M.: Scalar transport in random cylinder arrays at moderate Reynolds number. Journal of Fluid Mechanics 487(25), 43–79 (2003)
18. Worman, A., Kronnas, V.: Effect of pond shape and vegetation heterogeneity on flow and treatment performance of constructed wetlands. Journal of Hydrology 301(1-4), 123–138 (2005)
19. Wu, W.: Computational river dynamics. CRC (2007)

Impact of Different Recombination Methods in a Mutation-Specific MOEA for a Biochemical Application

Susanne Rosenthal, Nail El-Sourani, and Markus Borschbach

University of Applied Sciences, FHDW
Faculty of Computer Science, Chair of Optimized Systems,
Hauptstr. 2, D-51465 Bergisch Gladbach, Germany
{Susanne.Rosenthal,Nail.El-Sourani,Markus.Borschbach}@fhdw.de

Abstract. Peptides play a key role in the development of drug candidates and diagnostic interventions, respectively. The design of peptides is cost-intensive and difficult in general for several well-known reasons. Multi-objective evolutionary algorithms (MOEAs) introduce adequate in silico methods for finding optimal peptides sequences which optimize several molecular properties. A mutation-specific fast non-dominated sorting GA (termed MSNSGA-II) was especially designed for this purpose.

In this work, an empirical study is presented about the performance of MSNSGA-II which is extended by optionally three different recombination operators. The main idea is to gain an insight into the significance of recombination for the performance of MSNSGA-II in general - and to improve the performance with these intuitive recombination methods for biochemical optimization. The benchmark test for this study is a three-dimensional optimization problem, using fitness functions provided by the BioJava library.

Keywords: multi-objective biochemical optimization, character-encoded GA, recombination operators, MSNSGA-II.

1 Introduction

Multi-objective optimization (MOO) is an important issue in the fields of biochemistry, medicine and biology, especially for molecular modeling and drug-design methodologies. The aim of MOO in the chemoinformatic is to discover the optimal solution in a complex search space for several and often competing objective functions representing biological or chemical properties. In this field, multi-objective genetic algorithms are established tools [1] [2] [3] as they proved themselves as effective and robust methods. The most popular GA's are Pareto-based: NSGA[4] NSGA-II [5] SPEA [5], SPEA2 [7]. Generally, the GA procedure is based on optimization of populations of individuals which are improved in a multi-objective sense by the genetic components: recombination (or crossover), mutation and selection of a variation of the fittest individuals for

L. Vanneschi, W.S. Bush, and M. Giacobini (Eds.): EvoBIO 2013, LNCS 7833, pp. 188–199, 2013.

the next population. Variants of these components improve the convergence behavior and diversity significantly within the restriction and main challenge of a low number of generations. Especially the configuration and type of recombination and interaction with the mutation operator determines the search characteristic of a GA as they directly influence the 'broadness' of solution space traversal. There are a few theoretical results whether and how recombination operators in evolutionary algorithms can be applied most effectively, to improve convergence speed [8] [9] [10]. However, the suitable choice of components in MOEAs is an ongoing challenge; application specific and therefore usually based on empirical analysis. A detailed study of various recombination methods can be found in [11]. Recombination operators are usually adapted for the encoding of a GA. For real-coded GAs the recombination operators are categorized in mean-centric and parent-centric recombination approaches [23]. Here, unimodal normal distribution crossover (UNDX) [12], simplex (SPX) [13] and blend crossover (BLX) [14] are mean-centric operators, whereas the well-known simulated binary crossover (SBX) [15] used in NSGA-II and the parent-centric recombination operator (PCX) [11] are parent-centric approaches. The following recombination operators were used in a related work [3] - character-encoded GA: single point, double point, distance bisector, multi point, uniform and shuffle crossover. In each recombination, two parents are chosen randomly for recombination at randomly chosen points except for distance bisector crossover with a fixed central crossover point.

In this work, the character-encoded mutation-specific GA termed MSNSGA-II [16] is benchmarked, using three different recombination operators. The standard recombination method of MSNSGA-II varies the number of recombination points according to the Gaussian distribution [24]. Another recombination determines the number of recombination points according to a linear decreasing function. The third operator is a dynamic 2-point crossover, where the crossover points move to the edges of the individuals over the generations, while originating from the center. The performance of MSNSGA-II with the different recombination operators is benchmarked on a three-dimensional optimization problem: the three objective functions are taken from the BioJava library [20].

2 The Components of MSNSGA-II

MSNSGA-II was especially developed for biochemical applications. It was initially introduced in [16]. The fundamental idea for this GA was to improve upon the convergence behavior of the traditional NSGA-II (introduced by Deb et. al. [17]) and to customize NSGA-II for biochemical applications by exchanging the default component encoding, mutation, recombination and selection mechanisms with the ones specifically tailored for the proposed application. The genetic workflow of MSNSGA-II still corresponds to NSGA-II. The procedure of MSNSGA-II proceeds as follows:

Input: N Population size, T total number of generations

1. Randomly initialization of $P(t)$ with N individuals
2. Evaluation of objective functions
3. Pareto ranking and determination of crowding for each individual
4. Selection of parents for recombination and mutation
5. Evaluation of objective functions
6. Pareto ranking and determination of crowding distance
7. Determination of $P(t+1)$
8. Repeat steps 3.-7. if $t+1 < T$

The encoding applied was chosen to be intuitive for biochemical applications, in that the individuals are implemented as 20-character strings, each representing one of the natural canonical amino acids at each position. The remaining components differing to those of the standard NSGA-II are described in the following.

2.1 Recombination Operators

Every henceforth proposed recombination operator uses three randomly selected individuals as parents. The use of three parents for recombination proved to be suitable for a higher diversity within the solutions compared to the typical choice of two parents. The three parent recombination is based on the investigations of multi-parent recombination in evolutionary strategies of Eiben and Bäck [26]. Our empirical experiences reveal an overall low convergence if more predecessors (parents) are used, due to an enforced high exploration realized by high diversity.

The standard recombination operator of MSNSGA-II (described in [18]) uses a n-point recombination, referred to as 'Random'. The number n of recombination points varies according to a Gaussian distribution. Hence, the parameters of the recombination are the actuarial expectation $\rho = 2$ (the most frequent number for n) and the standard deviation $\sigma = 2.5$. Consequently, only positive values $n \geq 0$ are permitted: In the cases of negative values, the random generator is restarted.

A further recombination operator varies the number of recombination points over the generations via a linearly decreasing function:

$$x(t) = \frac{l}{2} - \frac{l/2}{T} \cdot t \tag{1}$$

which depends on the length of the individual l, the total number of generations the GA and t the actual generation. This operator is termed as 'LiDeRP' in the following. The basic idea of this recombination operator is that a higher number of recombination points in early generations leads to a broader search in the solution space, whereas a lower number in later generations supports the local search. The recombination points are chosen randomly.

The third recombination operator is a dynamic 2-point recombination where the recombination points move linearly to the edges of the sequence. The recombination points are determined by

$$p_1(t) = \frac{l}{2} - \frac{l/2}{T} \cdot t \tag{2}$$

$$p_2(t) = \frac{l}{2} + \frac{l/2}{T} \cdot t. \tag{3}$$

This operator is further termed '2-point-edges'.

2.2 Mutation Operators

In general, the mutation operators substitute certain characters with other ones chosen from a mutation pool comprised of 20 different characters representing amino acids. The first mutation operator is the deterministic dynamic mutation operator by Bäck and Schütz [19]. The mutation probabilities are determined by the function

$$p_{BS} = (a + \frac{l-2}{T-1}t)^{-1}, \tag{4}$$

with $a = 2$ and the corresponding variables mentioned in the last section. The mutation rate is bounded by $(0; \frac{1}{2}]$. In combination with LiDeRP as recombination, the function by Bäck and Schütz has been adapted to a lower initial mutation rate: $a = 4$ in (5).

The second mutation operator is termed as 'Random'. Here, the number of mutations is varied corresponding to the Gaussian distribution similar to the standard recombination. The parameters of this operator are the Gaussian interval length σ and the actuarial expectation ρ, which are set to: $\sigma = 5$ and $\rho = 0.2$.

The third operator is the self-adaptive scheme by Bäck and Schütz [19]. The mutation probabilities are determined via:

$$p_m(t+1) = (1 + \frac{1 - p_m(t)}{p_m(t)} \cdot e^{-\gamma N(0,1)})^{-1}, \tag{5}$$

where $N(0,1)$ is a normal distributed random number and the learning rate γ controls the adaption steps of the mutation rate. The traditional choice for the learning rate is $\gamma = 0.22$. The start population rate is set to $p_0 = 0.2$.

2.3 Selection Operator

The basic aggregation principles have been developed to combine different multi-objective goals solving the Rubics' cube [25] [27] - a problem of comparable solution space complexity. The following pseudo-code demonstrates the workings of the standard aggregate- selection-operator in MSNSGA-II. This operator was first introduced in [16].

1. select x (= tournament size) random individuals from population
2. Pareto-rank the tournament individuals
   (a) 0.5 probability
   (b) preselect front 0
   (c) randomly chose one to put into selection pool

   or opposite

   (a) 0.5 probability
   (b) SUS by front size to preselect individuals frome some front
   (c) randomly chose one to put into selection pool
3. if (selection pool size = μ) then done else back to 1.

The proposed selection method is a product of extensive analysis and careful iterative refinements. The specific design and parameter values are the product of an extensive empirical analysis and benchmarking process, using the proposed fitness functions. A more thorough and selection-focused examination of selection behavior under different GA-configurations such as different population and tournament sizes and optimization problems poses a highly interesting prospect and subject to ongoing work.

2.4 Fitness Functions

A simple three-dimensional minimization problem serves as benchmark test. The three objective functions were selected of the BioJava library: As the individuals are encoded as character strings, this enables the use of Needleman Wunsch algorithm (NMW) [21], Molecular Weight (MW) and *hydrophilicity* (hydro). A brief description of these functions is given in the following:

Needleman Wunsch Algorithm. The NMW algorithm performs a global sequence alignment of two sequences by using an iterative matrix method of calculation. More precisely, NMW as an objective function in MSNSGA-II is a measure for similarity of an individual to a pre-defined reference individual.

Molecular Weight. The fitness values for MW of an individual of the length l are determined from the individual amino acids (a_i) for $i = 1, ..., l$ [20]: Molecular weight is computed as the sum of mass of each amino acid plus a water molecule: $\sum_{i=1}^{l} mass(a_i) + 17.0073(OH) + 1.0079(H)$. (According to the periodic system of elements: Oxygen (O), hydrogen (H))

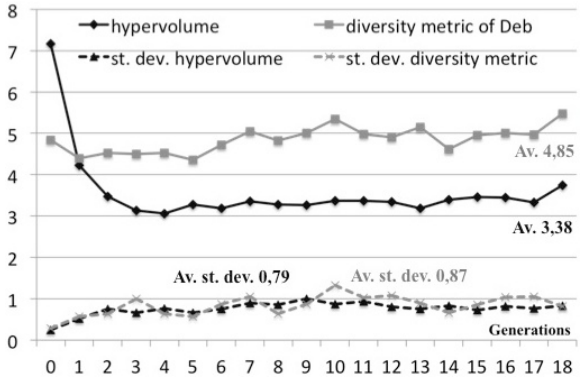
Hydrophilicity. The fitness value for hydrophilicity of an individual is calculated analogous to MW: $\frac{1}{l} \cdot (\sum_{i=1}^{l} hydro(a_i))$.
For a closer understanding and the source-code of these three objective functions, please refer to [20], [21].

3 Experiments

3.1 Simulation Onset and Metrics

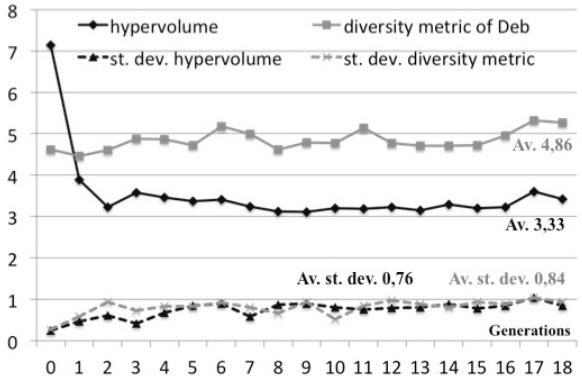
This section provides the results of the empirical studies. Here, each mutation operator is combined with each recombination operator to a configuration. The following parameter settings are the same for each configuration: The start population has a size of 100 randomly initialized individuals of 20 characters. Each configuration is run 30 times until the 18-th generation. This low number of generations is imposed by the first series of experiments [16] for the same three-dimension optimization problem - minimization of the objective functions NMW, MW and hydro. In these experiments, all MSNSGA-II configurations result in early convergence, meaning that an unusual low number (mainly under ten) is already sufficient to point out convergence behavior. Furthermore, MSNSGA-II exceeded the traditional NSGA-II for this optimization problem.

Mutation: deterministic dynamic operator by Bäck and Schütz



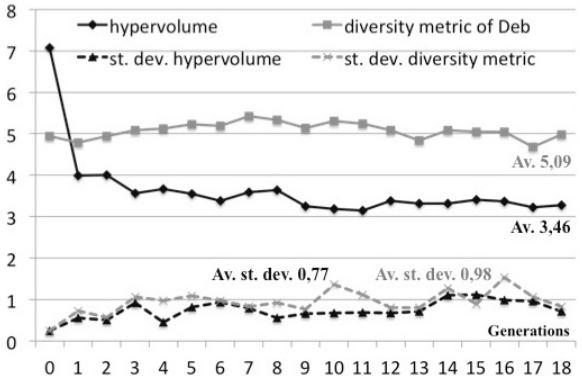
av.
confidence limits
hypervolume:
[2,99; 3,77]
diversity:
[4,41 ; 5,28]

Fig. 1. Recombination: Random



av.
confidence limits
hypervolume:
[2,95; 3,71]
diversity:
[4,44 ; 5,28]

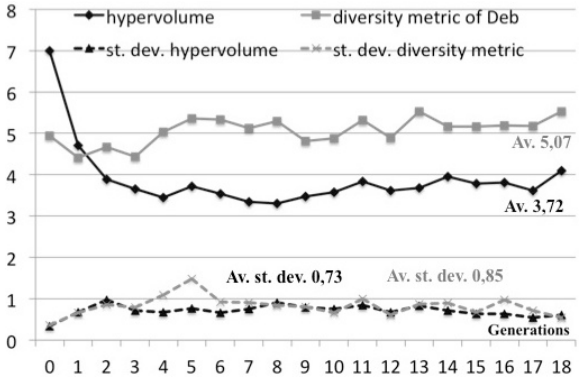
Fig. 2. Recombination: LiDeRP



av.
confidence limits
hypervolume:
[3,08; 3,84]
diversity:
[4,6 ; 5,58]

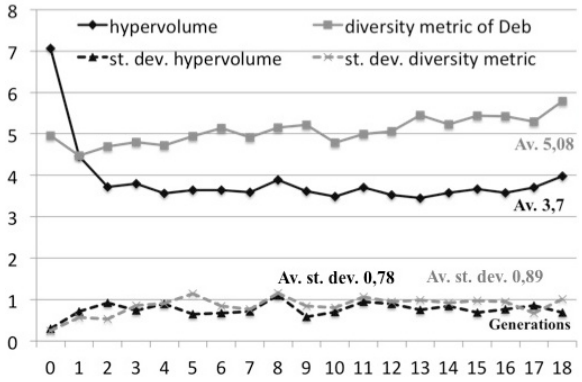
Fig. 3. Recombination: 2-point-edges

Mutation: Random



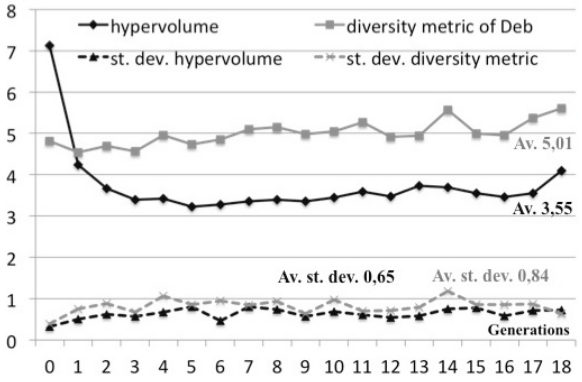
av.
confidence limits
hypervolume:
[3,36; 4,08]
diversity:
[4,65 ; 5,49]

Fig. 4. Recombination: Random



av.
confidence limits
hypervolume:
[3,31; 4,09]
diversity:
[4,64 ; 5,52]

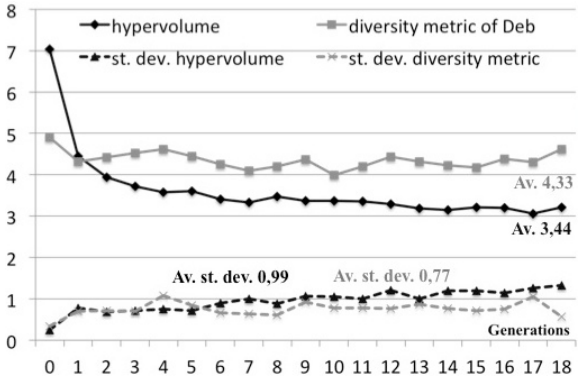
Fig. 5. Recombination: LiDeRP



av.
confidence limits
hypervolume:
[3,23; 3,87]
diversity:
[4,59 ; 5,43]

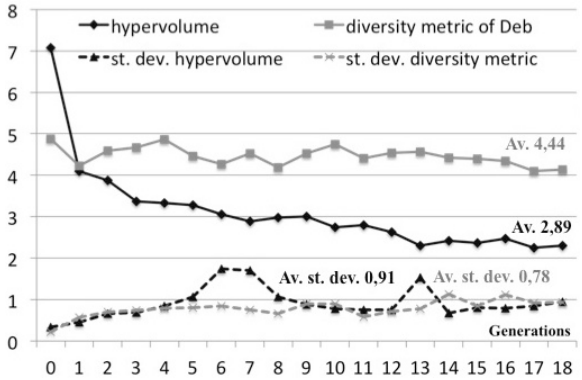
Fig. 6. Recombination: 2-point-edges

Mutation: self-Adaptive Operator by Bäck and Schütz



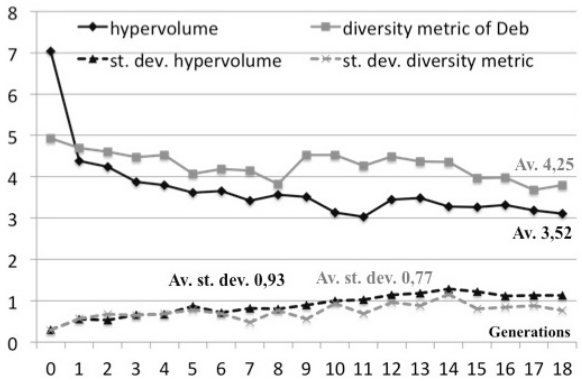
av.
confidence limits
hypervolume:
[2,95; 3,93]
diversity:
[3,95 ; 4,71]

Fig. 7. Recombination: Random



av.
confidence limits
hypervolume:
[2,44; 3,34]
diversity:
[4,05 ; 4,83]

Fig. 8. Recombination: LiDeRP



av.
confidence limits
hypervolume:
[3,06; 3,98]
diversity:
[3,87 ; 4,63]

Fig. 9. Recombination: 2-point-edges

Two metrics are consulted for evaluation: The S-metric or hypervolume of Zitzler [22] is used as a measure for convergence of the Pareto front. The hypervolume measures the space spanned by a set of non-dominated solutions to a pre-defined reference point - in this case **0**. Here, the hypervolume is determined of the whole population as we focus on the convergence behavior of the whole population to the optimal front. The diversity metric of Deb [17] is a measure for the diversity of the solution set. The aim of MOO is to receive optimal convergence behavior and well-spread non-dominated solution set at the same time. The average metric values as well as the standard deviations over 30 runs for each configuration are depicted in figure 1 to 9. The metric values are scaled. The objective function values reflect the distance of the individuals' objective function values to the one of a non-varying reference individual. The average metric values are tested according to statistical significance by one-tailed t-tests at a significant level of $0,05$: For the hypervolume an upper tailed t-test and for the diversity metric a lower-tailed t-test were performed. All average metric values received a clear statistical significance. Furthermore, beside Fig. 1 to 9 confidence limits are listed at the same significant level of $0,05$ each for the hypervolume and for the diversity metric.

3.2 Evaluation

An influence by variation of recombination operator is clearly visible, whereas the convergence behavior as well as the diversity within the solutions is mainly governed by the mutation operators. Therefore, the recombination operators are supposed to take the role to support the convergence behavior and diversity by its interaction with the mutation operator. The recombination operator LiDeRP reveals - both in combination with the deterministic dynamic mutation of Bäck and Schütz - and with the mutation Random slightly improved performances (Fig. 2 and Fig. 5) compared to the standard recombination Random in MSNSGA-II (Fig.1 and Fig. 4). A performance improvement is achieved when both a decrease of the hypervolume and an increase of the diversity can be identified. The recombination LiDeRP results in a clearly visible improved performance in combination with the self-adaptive mutation (Fig. 8) compared to the combination with recombination Random (Fig. 7). Merely the standard deviations of the configuration with the self-adaptive mutation exhibit an increasing in the hypervolume and a slight decreasing in the diversity.

The recombination 2-point-edges in combination with the self-adaptive mutation (Fig. 9) results in deterioration of the performance compared to the other two recombination operators (Fig. 7 and Fig. 8). The combination 2-point-edges and the deterministic dynamic mutation (Fig. 3) amount to diversity increase at the cost of the convergence when compared to both recombination Fig. 1 and 2.. However, an improvement in convergence is noticeable for the combination of 2-point-edges with the mutation Random (Fig. 6), only yielding a slight decrease of diversity compared to Fig. 4 and 5.

In general, the recombination LiDeRP achieved the best results independent of the chosen mutation operator.

4 Conclusion

This paper presents an empirical study about the performance of the character-encoded mutation-specific GA MSNSGA-II extended by three different recombination operators. The main interest of this study is to gain an insight of the recombination operators for the performance in general and to possibly improve the performance. The performance was tested on a synthetic three-dimensional optimization problem. The objective functions were drawn from the BioJava library and present an optimization problem on the subject of molecular features. Every possible combination of mutation- and recombination operators was inspected as a distinct GA configuration. It can be noted that the convergence behavior and the diversity within the solutions is mainly governed by the mutation operators. The recombination operators are supposed to take the role to support the convergence behavior and the diversity by its interaction with the mutation operators. The recombination operator LiDeRP combined with any mutation operator achieved the best result in convergence and diversity, especially with self-adaptive mutation. The recombination operator 2-point-edges combined with self-adaptive mutation showed disappointing results in convergence as well as in diversity. In combination with the deterministic dynamic mutation and mutation Random, the strides in diversity were made at the cost of convergence -and vice versa combined with Random.

As this approach seems promising for biochemical research, a closer look is necessary to gain a deeper inside of the convergence velocity rate. Future research will focus on the theoretical analysis of the interaction between mutation and recombination operators and the resulting performance. Feasible theoretical results will be utilized in the design and improvement of mutation and recombination variants. Furthermore, it will be examined how more functions -for instance from the BioJava library or other similar tools- can be used so that the results may be generalized further to real-life applications.

References

1. Vainio, M.J., Johnson, M.S.: Generating conformer ensembles using a multiobjective genetic algorithm. J. Chem. Inf. Model. 47(6), 2462–2474 (2007)
2. Nicolaou, C.A., Brown, N., Pattichis, C.S.: Molecular optimization using computational multi-objective methods. Drug Discovery & Development 10(3), 316–324 (2007)
3. Knapp, B., Gicziv, V., Ribarics, R.: PeptX: Using genetic algorithms to optimize peptides for MHC binding. BMC Bioinformatics 12, 241 (2011)
4. Srinivas, N., Deb, K.: Multiobjective optimization using nondominated sorting in genetic algorithms. J. Evol. Comput. 2(3), 221–248 (1994)
5. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. 6(2), 182–197 (2002)

6. Zitzler, E., Thiele, L.: An evolutionary algorithm for multiobjective optimization: The strength Pareto approach. Technical report 43, Computer engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology, ETH (1999)
7. Zitzler, E., Laumann, M., Thiele, L.: Improving the strength pareto evolutionary algorithm. Technical report 103, Computer engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich (2001)
8. Jansen, T., Wegener, I.: Real royal road functions - where crossover probably is essential. Discrete Applied Mathematics 149(1-3), 111–125 (2005)
9. Jansen, T., Wegener, I.: The analysis of evolutionary algorithms - a proof that crossover really can help. Algorithmica 34(1), 47–66 (2002)
10. Neumann, F., Theile, M.: How Crossover Speeds Up Evolutionary Algorithms for the Multi-criteria All-Pairs-Shortest-Path Problem. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN XI, Part I. LNCS, vol. 6238, pp. 667–676. Springer, Heidelberg (2010)
11. Deb, K., Anand, A., Joshi, D.: A computationally Efficient Evolutionary Algorithm for Real Parameter Optimization, KanGAL report: 2002003
12. Ono, I., Kobayashi, S.: A real-coded genetic algorithm for functional optimization using unimodal normal distribution crossover. In: Proceedings of the 7th International Conference on Genetic Algorithms (ICGA-7), pp. 246–253 (1997)
13. Tsusui, S., Yamamura, M., Higuchi, T.: Multi-parent recombination with simplex crossover in real-coded genetic algorithms. In: Proceedings of the Genetic and Evolutionary Computing Conference (GECCO 1999), pp. 657–664 (1999)
14. Eshelman, L.J., Schaffer, J.D.: Real-coded genetic algorithms and interval schemata. In: Whitley, D. (ed.) Foundation of Genetic Algorithm II, pp. 187–202 (1993)
15. Deb, K., Agrawal, R.B.: Simulated binary crossover for continuous search space. Complex System 9, 115–148 (1995)
16. Rosenthal, S., El-Sourani, N., Borschbach, M.: Introduction of a Mutation Specific Fast Non-dominated Sorting GA Evolved for Biochemical Optimizations. In: Bui, L.T., Ong, Y.S., Hoai, N.X., Ishibuchi, H., Suganthan, P.N. (eds.) SEAL 2012. LNCS, vol. 7673, pp. 158–167. Springer, Heidelberg (2012)
17. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. 6(2), 182–197 (2002)
18. Borschbach, M.: Neural classification of biological properties and genetic operators configuration issues. Trans. on Information Science 12(2), 324–329 (2005) ISSN: 1790-0832
19. Bäck, T., Schütz, M.: Intelligent Mutation Rate Control in Canonical Genetic Algorithms. In: Michalewicz, M., Raś, Z.W. (eds.) ISMIS 1996. LNCS, vol. 1079, pp. 158–167. Springer, Heidelberg (1996)
20. BioJava: CookBook, release 3.0, `http://www.biojava.org/wiki/BioJava`
21. Needleman, S.B., Wunsch, C.D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology 48(3), 443–453 (1970)
22. Zitzler, E., Thiele, L.: Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) PPSN 1998. LNCS, vol. 1498, pp. 292–301. Springer, Heidelberg (1998)
23. Deb, K., Joshi, D., Anand, A.: Real-coded evolutionary algorithms with parent-centric recombination. KanGAL Report No. 2001003 (2001)
24. Röckendorf, N., Borschbach, M., Frey, A.: Molecular evolution of peptide ligands with custom-tailored characteristics. PLOS Computational Biology (December 2012), open access journal

25. El-Sourani, N., Borschbach, M.: Design and Comparison of two Evolutionary Approaches for Solving the Rubik's Cube. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN XI, Part II. LNCS, vol. 6239, pp. 442–451. Springer, Heidelberg (2010)
26. Eiben, A.E., Bäck, T.: Empirical investigation of multiparent recombination operators in evolutionary strategies. Evolutionary Computation 5(3), 347–365 (1997)
27. Borschbach, M., Grelle, C., Hauke, S.: Divide and Evolve Driven by Human Strategies. In: Deb, K., Bhattacharya, A., Chakraborti, N., Chakroborty, P., Das, S., Dutta, J., Gupta, S.K., Jain, A., Aggarwal, V., Branke, J., Louis, S.J., Tan, K.C. (eds.) SEAL 2010. LNCS, vol. 6457, pp. 369–373. Springer, Heidelberg (2010)

Cell-Based Metrics Improve the Detection of Gene-Gene Interactions Using Multifactor Dimensionality Reduction

Jonathan M. Fisher[1], Peter Andrews[1], Jeff Kiralis[1], Nicholas A. Sinnott-Armstrong[1], and Jason H. Moore[1,2,3]

[1] Department of Genetics, Geisel School of Medicine, Dartmouth College, Hanover, NH 03755, USA
[2] Department of Community and Family Medicine, Geisel School of Medicine, Dartmouth College, Hanover, NH 03755, USA
[3] Institute for Quantitative Biomedical Sciences, Dartmouth College, Hanover, NH 03755, USA
jonathan.m.fisher@dartmouth.edu
http://www.epistasis.org/

Abstract. Multifactor Dimensionality Reduction (MDR) is a widely-used data-mining method for detecting and interpreting epistatic effects that do not display significant main effects. MDR produces a reduced-dimensionality representation of a dataset which classifies multi-locus genotypes into either high- or low-risk groups. The weighted fraction of cases and controls correctly labelled by this classification, the balanced accuracy, is typically used as a metric to select the best or most-fit model. We propose two new metrics for MDR to use in evaluating models, Variance and Fisher, and compare those metrics to two previously-used MDR metrics, Balanced Accuracy and Normalized Mutual Information. We find that the proposed metrics consistently outperform the existing metrics across a variety of scenarios.

Keywords: Multifactor Dimensionality Reduction; Fisher's exact test.

1 Introduction

Epistasis, or gene-gene interaction, is fundamental in gene expression, and figures prominently in the genetics of complex traits such as susceptibility to disease (e.g., [3,13]). Epistasis introduces complexity in the relationship between genotype and phenotype, making patterns in that relationship more difficult to detect. We therefore need tools that enable us to detect epistasis and search for the patterns that might be hidden behind it [20,8,9,19,21,13,15]. In particular, our tools should make use of all of the information available in each dataset.

Multifactor Dimensionality Reduction (MDR) is a non-parametric data-mining tool that can detect epistatic models of gene expression that do not show significant main effects, widely used in the study of genetic traits with or without a component of environmental causation [18,17,6,5,7,10,11,24]. MDR

L. Vanneschi, W.S. Bush, and M. Giacobini (Eds.): EvoBIO 2013, LNCS 7833, pp. 200–211, 2013.

uses a constructive-induction algorithm to label each genotype combination as high-risk or low-risk based on a discrete endpoint such as case–control status, constructing a new variable with two risk levels which pools all high-risk genotypes into one group and all low-risk genotypes into another group[11,10]. That new variable can then be analyzed with a classification method such as naïve Bayes or logistic regression.

For any desired order of interaction N (typically between 2 and 4), MDR iterates over all sets of N loci and constructs a model for each of them. Each individual in the dataset is classified according to which allele it has at each locus of the model, and a case-control table is constructed which counts how many individuals of each allelic combination are cases and how many are controls (Fig. 1B). That case-control table encapsulates the model for that set of loci. MDR then chooses, from all of the models, the case-control table that scores highest by whatever metric (i.e., measure of model fitness) it uses. Finally, MDR constructs a new variable from that case-control table, labeling each combination of genotype values as low-risk if the corresponding cell in the case-control table has a ratio of cases to controls below a pre-chosen threshold, and high-risk otherwise.

In standard MDR, the metric used is balanced accuracy [24], defined as the mean of sensitivity and specificity:

$$\frac{TP/(TP+FN)+TN/(TN+FP)}{2}, \tag{1}$$

where TP is the number of true positives, TN the number of true negatives, FP the number of false positives, and FN the number of false negatives. The threshold typically used is the ratio of cases to controls in the dataset as a whole, as recommended by Velez et al [24]. Various alternative metrics to balanced accuracy have been investigated, but they have typically focused on evaluating each possible model simply in terms of the contingency table of the number of true positives, true negatives, false negatives, and false positives that the model produces on the given dataset [1].

We propose to improve the metric used by MDR by making full use of the information available in the dataset – looking at the full case-control table of status versus genotype, instead of the summary table of risk-level versus status. Specifically, we propose two new metrics, one based on a Fisher's exact test applied to the case-control table, and one based on the variance of case-control ratios in the table. We evaluate the metrics for their ability to pick out a known signal from noise in simulated datasets across a wide variety of scenarios, and compare their performance to that of standard metrics. We show that the new metrics display equal or greater detection ability across all of the scenarios investigated, with significantly improved ability to detect weaker signals.

2 Methods

2.1 Use of Metrics

The function of MDR is to construct a new attribute by selecting the model which, for a given dataset, best predicts the phenotype from the genotypic information.

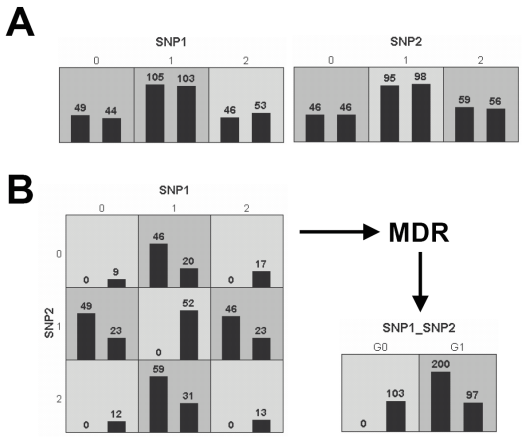


Fig. 1. MDR attribute construction. (A) illustrates the distribution of cases (left bars) and controls (right bars) for each of the three genotypes of SNP1 and SNP2 in an example dataset. The shading of the cells indicates the labeling that has been assigned (using a threshold of T = 1): dark shading indicates "high-risk" and light shading "low risk". (B) illustrates the distribution of cases and controls when the two functional SNPs are considered jointly. A new single attribute is constructed by pooling the high-risk genotype combinations into one group, G1, and the low-risk into another group, G0. Reprinted from Velez et al [24].

Each model is based on a small number of interacting loci, typically between 1 and 4, and yields the case-control table of status versus genotype for those loci over the dataset. MDR works in two phases, first selecting a model within each level of interaction, and then choosing a model from among the levels. Within a given level of interaction, MDR uses a metric to score each case-control table produced from the dataset and then selects the case-control table with the highest score assigned by the metric. We concentrate here on evaluating the metrics used to compare models within a single level of interaction.

The value of each metric on a model is calculated by first constructing the case-control table of the model over the dataset and then applying some formula to the table; the exact nature of the formula is what defines the metric.

2.2 Definitions of Metrics

We propose two new metrics for MDR, Variance and Fisher; we evaluate them by comparing them to two metrics that have been used previously in MDR: Balanced Accuracy [24], which is the standard metric used in MDR, and Normalized Mutual Information, which has been recommended by Bush et al [1]. In each evaluation-run we test the detection ability of each metric, the ability to identify the set of loci that constitute the known signal.

The Variance metric is motivated by the concept of the variance of the case-fractions in a case-control table: for a table with N cells, the Variance metric is defined as the total variance of case-fractions in the table,

$$\sum_{i=1}^{N} p_i(k_i - k)^2 \ , \tag{2}$$

where N is the number of cells in the table, p_i is the fraction of the individuals that lie in the ith cell (which is the sample approximation to the probability that an individual chosen at random lies in the ith cell), k_i is the fraction of individuals in the ith cell that have the condition or trait (i.e., the case-fraction in the cell), and k is the fraction of individuals that have the condition or trait in the dataset as a whole. If the the ith cell cell is empty, we define the ith term of the sum to be 0.

The Fisher metric uses a Fisher's exact test to measure how unusual each cell of a case-control table is, by looking at the numbers of cases and controls in each cell and calculating the probability of getting case-control values which are at least as skewed as the observed case-control values; the per-cell probabilities are combined to give an approximate log-probability of a given table. Intuitively, the lower the probability of a given case-control table arising by chance, the more likely it is to represent an underlying biological phenomenon. Given a dataset with a total of A cases and B controls, if cell i of a case-control table has a cases and b controls, we set T_i equal to the value of Fisher's exact test applied to

$$\begin{pmatrix} a & A-a \\ b & B-b \end{pmatrix}. \tag{3}$$

Thus T_i is then the two-tailed p-value for selecting a cases and b controls by chance from a total of A cases and B controls. Note that if the ith cell is empty then a and b are 0 and T_i is 1. Then, using Fisher's method [4] to combine the probabilities, the Fisher metric over the whole table is defined as

$$\sum_{i=1}^{N} -2\log(T_i) \ , \tag{4}$$

where N is the number of cells in the table. Note that, because the cells of a case-control table are not independent, the value of the Fisher metric for a table will not correspond to an exact probability; however, calculating an exact probability would be prohibitively expensive, and we hypothesize that the approximate probability used in the Fisher metric will be an effective method of scoring case-control tables.

For our comparison of metrics, we implemented Balanced Accuracy and Normalized Mutual Information, both of which are based on the risk-vs-status contingency table of true positives (TN), false negatives (FN), true negatives (TN), and false positives (FP). Balanced Accuracy, as described by Velez et al [24], is the mean of sensitivity and specificity, as shown in Eqn. 1 above, and the

corresponding metric selects the model with the highest balanced accuracy. In Normalized Mutual Information, as described by Bush et al [1], three entropies are calculated from the risk-vs-status contingency table: the row entropy, the column entropy, and a conditional entropy:

$$H(x) = -\sum_i p_i \log_2 p_i \ , \tag{5}$$

$$H(y) = -\sum_j p_j \log_2 p_j \ , \tag{6}$$

$$H(y|x) = -\sum_i p_i \sum_i \frac{p_{ij}}{p_j} \log_2 \frac{p_{ij}}{p_j} \ . \tag{7}$$

The quantity p_j is the empirical probabilities of being a case, p_i is the empirical probability of being high-risk, and p_{ij} is their joint probability. Using these entropy values, Normalized Mutual Information (NMI) is defined as:

$$NMI(y) = \frac{H(y) - H(y|x)}{H(y)}. \tag{8}$$

The Normalized Mutual Information metric selects the model with the highest value.

2.3 Numerical Analysis

We evaluated each of the four metrics over numerous different scenarios, and compared the abilities of the four metrics to distinguish a specified signal from noise (defined as a given metric assigning a higher score to the signal model than to each of the other models in the iteration). We did this by running MDR on two collections of simulated datasets: the datasets used by Velez et al [24], and a new, more comprehensive, collection of datasets generated by the GAMETES software [23,22].

In the first set of tests, we used the datasets described in Velez et al [24], which are based on a set of 70 models. Those models are based on two-way epistatic interactions with no main effect, use two minor-allele frequencies, 0.2 and 0.4, and range over the heritabilities 0.01, 0.025, 0.05, 0.1, 0.2, 0.3, and 0.4, giving a total of 14 parameter-pairs; for each of those parameter-pairs there are 5 distinct models, for a total of 70 models. Each of those models was used to generate 100 balanced datasets of 200, 400, 800, and 1600 total individuals; each dataset has 18 noise loci in addition to the 2 signal loci, for a total of 20 loci. Thus there are 7,000 datasets of each of the four sizes, for a total of 28,000 datasets. We ran MDR on each of those datasets, using each of the four metrics in turn, and evaluated how often each metric distinguished the signal model from the noise models.

In the second set of tests we tested how the ability of each metric to pick out the signal model varied with varying heritability, minor-allele frequency, and prevalence of the signal, and individual-count of the dataset overall. First, we generated 5,000 datasets for each of the heritabilities 0.01, 0.025, 0.05, 0.1, 0.2,

0.3, and 0.4; for each of those datasets, the minor-allele frequency was allowed to vary stochastically and uniformly between 0.1 and 0.5, the penetrance between 0.2 and 0.5, and the average number of individuals per cell of the case-control table between 10 and 80. Next, we generated 5,000 datasets for each of the minor-allele frequencies 0.1, 0.2, 0.3, 0.4, and 0.5; the heritabilities were allowed to vary between 0.01 and 0.4, and the penetrance and the number of individuals per cell were varied as before. Next, we generated 5,000 datasets for each of the penetrances 0.2 and 0.5, with the other parameters varying as before. Finally, we generated 5,000 datasets for each of the average individual-counts per cell 10, 20, 40, and 80, with the other parameters varying as before. We did all of these tests with a 2-locus signal, a 3-locus signal, and a 4-locus signal; in each case sufficient noise SNPs were added to achieve a total of 20 SNPs in each dataset.

The signal tables were generated using the GAMETES software[23,22]; however, there are limitations on the achievable heritabilities for high locus-counts and low minor-allele frequencies, manifested in GAMETES. Due to this limitation, for the datasets with a 3-locus signal the heritability was restricted to 0.3 or less and the minor-allele frequency to 0.2 or greater, and for the datasets with a 4-locus signal the heritability was restricted to 0.1 or less and the minor-allele frequency to 0.3 or greater.

As described above, we tested a variety of different dataset sizes. In order to make the results more comparable across different numbers of loci, we specified the dataset sizes in terms of average number of individuals per table-cell, instead of in terms of total number of individuals in the dataset, using average individual-counts per cell of 10, 20, 40, and 80 individuals. Thus, for example, we generated a 2-locus table (which has 9 cells; see Fig. 1) with a total of 90 individuals, and a 3-locus table (which has 27 cells) with a total of 270 individuals. Both tables then had an average of 10 individuals per cell. By keeping fixed the average number of individuals per cell instead of the total number of individuals in the dataset, we achieved the same degree of sparseness between the 2-locus table and the 3-locus table. If, instead, we had generated the 3-locus table with a total of just 90 individuals, its cells would be much sparser than the corresponding 2-locus table. Since the overall sparseness of individuals per cell can affect the behavior of the metrics, keeping the average number of individuals per cell constant across locus-counts makes the behavior of the metrics more comparable. Note that the goal of this approach is to maintain roughly comparable degrees of sparseness across the different levels of interaction, and we evaluate the metrics over a range of sparseness values. (With either approach to defining dataset sizes, by average-individual-count-per-cell or by total dataset size, the results will not be perfectly comparable across different numbers of interacting loci – but the central focus of this paper is to compare different metrics on similar scenarios, not across different scenarios.)

We also demonstrated the proposed new metrics by applying them to a population-based genetic study of tuberculosis (TB) that was previously analyzed using MDR by Collins et al.[2] The study analyzed 321 pulmonary TB cases and 347 healthy controls genotyped at The Bandim Health Project in Guinea Bissau.[14] Each individual was genotyped for 19 single-nucleotide

polymorphisms (SNPs) from immunological candidate genes VDR, DC-SIGN, PTX3, TLR2, TLR4, and TLR9. Collins et al imputed missing data using a frequency-based imputation and then filtered the dataset to six SNPs using ReliefF. They then applied MDR, which returned an overall best model consisting of SNPs rs2305619, rs187084, and rs1145421. In the present study, we applied all four metrics to the same filtered dataset.

Finally, we ran benchmark tests to evaluate the computation time for each metric. For each of 2-locus, 3-locus, and 4-locus interactions, we generated 10,000,000 random tables and scored each of the tables by each metric, recording the time to score each set of tables.

3 Results

3.1 Testing on the Velez Datasets

In the results of running MDR on the Velez et al[24] datasets, we group together the set of 5 models on each parameter-triple {minor-allele frequency, dataset size, heritability}, averaging the detection scores of each group. For each metric, "detection" is defined as the fraction of runs in which that metric assigned a higher score to the signal model than to each of the other models in the iteration. In every case the result was that both Variance and Fisher did as well as or better than both Normalized Mutual Information and Balanced Accuracy, with one exception: for 200-individual datasets with a minor-allele frequency of 0.2 and heritability of 0.01, Fisher had a detection score of 3.2% and Balanced Accuracy had a detection score of 3.4%. To get a high-level comparison between the various metrics, we took the overall average detection score for each metric, excluding those parameter-triples where all four metrics had detections of 0% or all four metrics had detections of 100%. By excluding the detection scores of the scenarios where either all metrics always failed or all metrics always succeeded, we concentrate on situations where the metrics differ in their effectiveness. As seen in Table 1, Variance and Fisher did about 4 to 4.5 percentage-points better than Normalized Mutual Information and Balanced Accuracy by this measure.

In order to quantify the degree to which the Variance and Fisher metrics improved over the Normalized Mutual Information and Balanced Accuracy metrics, for each parameter-triple {minor-allele frequency, dataset size, heritability} we calculated the χ^2 statistic between: the Variance metric and the Normalized

Table 1. Average detection abilities on the Velez datasets, where detection is not 0% or 100%

Metric	Detection
Variance	73.5%
Fisher	73.0%
Normalized Mutual Information	68.9%
Balanced Accuracy	69.0%

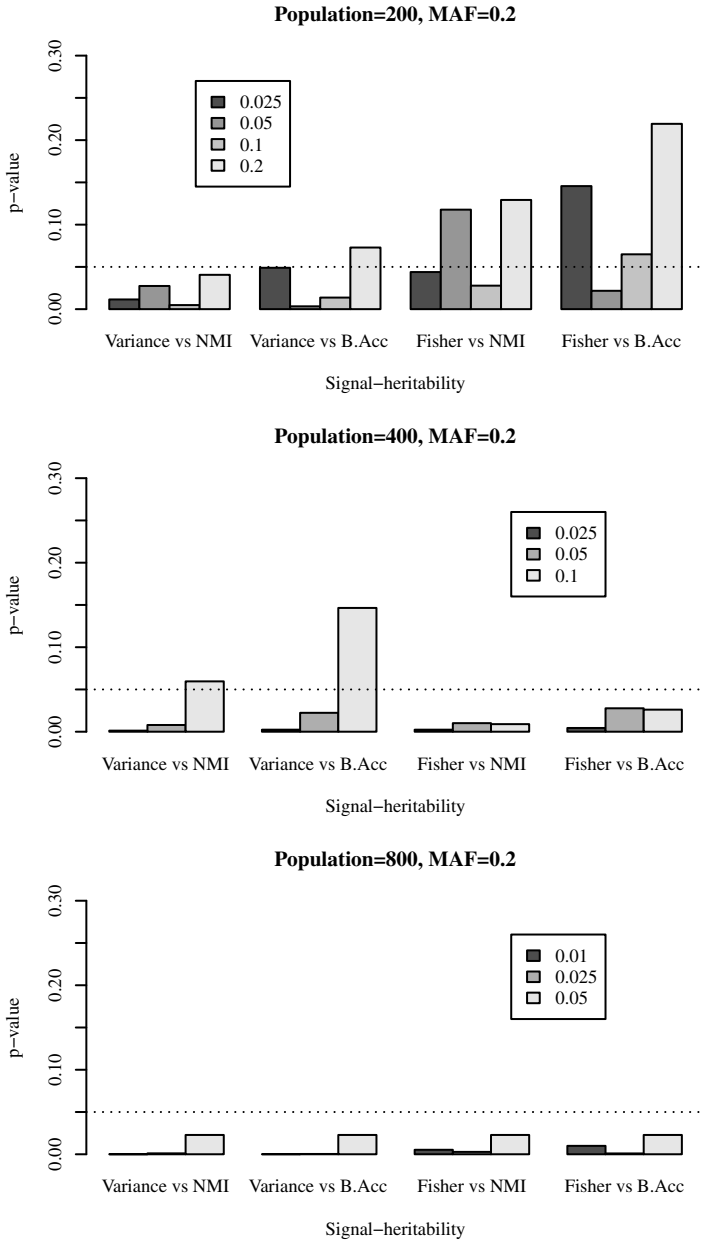


Fig. 2. Significance of χ^2 statistics for the comparisons between the Variance and Fisher metrics and the Normalized Mutual Information (NMI) and Balanced Accuracy (B.Acc) metrics, for the signal-heritabilities listed in the legends

Mutual Information metric; Variance and Balanced Accuracy; Fisher and Normalized Mutual Information; and Fisher and Balanced Accuracy. For example, for the χ^2 statistic between the Variance metric and the Normalized Mutual Information metric for a given scenario, we constructed a table of the success and failure counts for each of the metrics under that scenario, and calculated the R chisq.test function on that table. We then calculated the statistical significance of each χ^2 statistic. We show selected results as follows; in the results not shown, differences between the metrics were usually marginally significant or not significant. As mentioned above, in all scenarios where there was a significant difference, Variance and Fisher scored better than the Normalized Mutual Information and Balanced Accuracy metrics.

For 200-individual datasets and a minor-allele frequency of 0.2, the improvement of the Variance metric over the Normalized Mutual Information metric was significant at the 0.05 level or better for heritabilities of 0.025, 0.05, 0.1, and 0.2 (Fig. 2). The improvement of the Variance metric over the Balanced Accuracy metric was very significant for heritabilities of 0.05, and 0.1 and marginally signficant for heritabilities of 0.025 and 0.2. The comparison between the Fisher metric and the Normalized Mutual Information and Balanced Accuracy metrics was mixed, as seen in the figure.

For 400-individual datasets and a minor-allele frequency of 0.2, the improvement of the Variance and Fisher metrics over the Normalized Mutual Information and Balanced Accuracy metrics was very significant for heritabilities of 0.025, 0.05, and 0.1, except for the Variance metric in the scenario with heritability of 0.1.

For 800-individual datasets and a minor-allele frequency of 0.2, the improvement of the Variance and Fisher metrics over the Normalized Mutual Information and Balanced Accuracy metrics was very significant for heritabilities of 0.01, 0.025, and 0.05.

3.2 Comprehensive Testing on GAMETES Datasets

In the second part of the study we tested the effectiveness of each metric over a wide range of scenarios; we found that the two new metrics, Variance and Fisher, always did as well as or better than Normalized Mutual Information and Balanced Accuracy, with one exception: in the scenario of a 2-way signal with heritability of 0.4, the Normalized Mutual Information and Balanced Accuracy metrics score 99.98% while the Variance metric scored 99.96%; in that scenario the Fisher metric scored 100%. The only scenarios where the two older metrics were close to the two new metrics were scenarios where all four metrics had scores near 100%, usually because the heritability of the signal was high, making it easy to detect. Whenever the metrics scored less than 85%, the two new metrics outscored the two older metrics by at least two percentage points. Thus we see that when the signal is relatively easy to find the two new metrics do as well as or better than the older metrics, and when the signal is harder to detect the new metrics do significantly better – by ten percentage points or more in five of the scenarios.

These observations are made more precise by using a χ^2 analysis. As in the analysis of the Velez datasets, we calculated the χ^2 statistic between: the

Variance metric and the Normalized Mutual Information metric; Variance and Balanced Accuracy; Fisher and Normalized Mutual Information; and Fisher and Balanced Accuracy. We then calculated the statistical significance of each χ^2 statistic. In most scenarios, the improvement of the new metrics over the older ones is highly significant. We found three categories of performance: When all four metrics detect the signal correctly 99% of the time or more, the χ^2 comparisons between the metrics showed no significant difference, with p-values of 0.3 or greater. When the metrics had a correct detection level between 85% and 99%, the χ^2 comparisons showed the new metrics significantly better than the older metrics, with p-values between 0.04 and 0.001. And when the signal was harder to detect, with the metrics finding the correct signal less than 85% of the time, the improvement of the new metrics over the older ones was highly significant, with p-values less than 0.001.

In the comprehensive testing using GAMETES datasets there were 5,000 datasets for each scenario, as compared with the 500 datasets per scenario in the Velez datasets; we see that with the greater resolution afforded by the larger number of datasets, the improvement inherent in the new metrics becomes crystal clear.

3.3 Demonstration Data and Benchmarks

We also tested the four metrics on a tuberculosis dataset that had previously been evaluated using MDR, which found an overall best model consisting of SNPs rs2305619, rs187084, and rs1145421. In our tests, all four of the metrics identified that model as best overall.

The computation times for the Variance, Normalized Mutual Information, and Balanced Accuracy metrics are similar (Tbl. 2). The Fisher metric, being more computationally intensive, takes considerably longer to run; however, that run-time could be improved dramatically by caching the probability calculations.

Table 2. Time in seconds to calculate each metric on 10,000,000 tables for 2-locus to 4-locus interactions, running in Java on a 2.26 GHz Intel Xeon with single-threading

Metric	2-locus	3-locus	4-locus
Variance	2.81	6.87	15.84
Fisher	138.99	248.56	593.33
Normalized Mutual Information	5.6	8.08	13.95
Balanced Accuracy	2.33	6.84	21.14

4 Discussion

The ability to discover the connections between genotype and phenotype is central to genomics research, but it continues to be challenging. It was over a decade ago that Risch and Merikangas first seriously proposed the testing of all known

SNPs in the human genome for disease association either directly or by LD with other SNPs [16]. Today, it is becoming cost effective to measure a million SNPs with widely-available human SNP arrays, but the tools used to analyze that data need to improve as well [12]. Part of that improvement is to ensure that all of the information contained within each dataset is fully employed.

The reduced-dimensionality high-risk/low-risk contingency table produced by MDR contains less information than the case-control table for the model it represents – each cell of the case-control table contains information about the numbers of cases and controls in that cell, and that information is omitted when the cases and controls are summed into the contingency table. Thus, by defining our metrics, for the purpose of model-selection only, directly on the case-control tables instead of on the contingency tables, we are able to make better use of that information in selecting a model. Once selected, the model is reduced to a high-risk/low-risk contingency table in the usual way.

We find that the Variance and Fisher metrics do as well as or significantly better than Normalized Mutual Information and Balanced Accuracy in all of the wide variety scenarios in which they were tested, as measured in terms of detection ability. The improvement is especially strong when the signal is difficult to detect, which is exactly the scenario where improvement is most desirable. The Fisher metric is of particular value because it gives a direct measure of how unlikely a given model is to have arisen by chance, and therefore of how likely the model is to reflect an underlying biological phenomenon. However, it takes substantially more computation time than any of the other metrics tested. Given that the Variance metric closely parallels the Fisher metric in all regimes tested, we recommend the Variance metric for use with MDR going forward.

Acknowledgments. This work was supported by NIH grants LM009012, LM010098, and AI59694.

References

1. Bush, W.S., Edwards, T., Dudek, S., McKinney, B., Ritchie, M.: Alternative contingency table measures improve the power and detection of multifactor dimensionality reduction. BMC Bioinformatics 9, 238 (2008)
2. Collins, R.L., Hu, T., Wejse, C., Sirugo, G., Williams, S., Moore, J.: Multifactor dimensionality reduction reveals a three-locus epistatic interaction associated with susceptibility to pulmonary tuberculosis (2012) (manuscript submitted for publication)
3. Cordell, H.: Epistasis: what it means, what it doesn't mean, and statistical methods to detect it in humans. Hum. Mol. Genet. 11, 2463–2468 (2002)
4. Fisher, R.: Statistical methods for research workers. Genesis Publishing Pvt. Ltd. (1925)
5. Hahn, L., Moore, J.: Ideal discrimination of discrete clinical endpoints using multilocus genotypes. In Silico Biol. 4, 0016 (2004)
6. Hahn, L., Ritchie, M., Moore, J.: Multifactor dimensionality reduction software for detecting gene-gene and gene-environment interactions. Bioinformatics 19, 376–382 (2003)

7. Moore, J.H.: Computational analysis of gene-gene interactions in common human diseases using multifactor dimensionality reduction. Expert Rev. Mol. Diagn. 4, 795–803 (2004)
8. Moore, J.: The ubiquitous nature of epistasis in determining susceptibility to common human diseases. Hum. Hered. 56, 73–82 (2003)
9. Moore, J.: A global view of epistasis. Nat. Genet. 37, 13–14 (2005)
10. Moore, J.: Genome-wide analysis of epistasis using multifactor dimensionality reduction: feature selection and construction in the domain of human genetics. In: Zhu, X., Davidson, I. (eds.) Knowledge Discovery and Data Mining: Challenges and Realities with Real World Data, pp. 17–30. IGI Press, Hershey (2007)
11. Moore, J., Gilbert, J., Tsai, C., Chiang, F., Holden, W., Barney, N., White, B.: A flexible computational framework for detecting, characterizing, and interpreting statistical patterns of epistasis in genetic studies of human disease susceptibility. J. Theor. Biol. 241, 252–261 (2006)
12. Moore, J., Williams, S.: New strategies for identifying gene-gene interactions in hypertension. Ann. Med. 34, 88–95 (2002)
13. Moore, J., Williams, S.: Traversing the conceptual divide between biological and statistical epistasis: systems biology and a more modern synthesis. BioEssays 27, 637–646 (2005)
14. Olesen, R., Wejse, C., Velez, D., Bisseye, C., Sodemann, M., Aaby, P., Rabna, P., Worwui, A., Chapman, H., Diatta, M., Adegbola, R., Hill, P., Stergaard, L., Williams, S., Sirugo, G.: Dc-sign (cd209), pentraxin 3 and vitamin d receptor gene variants associate with pulmonary tuberculosis risk in West Africans. Genes and Immunity 8(suppl. 6), 456–467 (2007)
15. Rea, T., Brown, C., Sing, C.: Complex adaptive system models and the genetic analysis of plasma hdl-cholesterol concentration. Perspect. Biol. Med. 49, 490–503 (2006)
16. Risch, N., Merikangas, K.: The future of genetic studies of complex human disease. Science 273, 1516–1517 (1996)
17. Ritchie, M., Hahn, L., Moore, J.: Power of multifactor dimensionality reduction for detecting gene-gene interactions in the presence of genotyping error, missing data, phenocopy, and genetic heterogeneity. Genet. Epidemiol. 24, 150–157 (2003)
18. Ritchie, M., Hahn, L., Roodi, N., Bailey, L., Dupont, W., Parl, F., Moore, J.: Multifactor dimensionality reduction reveals high-order interactions among estrogen metabolism genes in sporadic breast cancer. Am. J. Hum. Genet. 69, 138–147 (2001)
19. Sing, C., Stengard, J., Kardia, S.: Genes, environment, and cardiovascular disease. Arterioscler. Thromb. Vasc. Biol. 23, 1190–1196 (2003)
20. Templeton, A.: Epistasis and complex traits. In: Wade, M., Brodie III, B., Wolf, J. (eds.) Epistasis and Evolutionary Process. Oxford University Press, New York (2000)
21. Thornton-Wells, T., Moore, J., Haines, J.: Genetics, statistics, and human disease: analytical retooling for complexity. Trends Genet. 20, 640–647 (2004)
22. Urbanowicz, R., Kiralis, J., Fisher, J., Moore, J.: Predicting the difficulty of pure, strict, epistatic models: metrics for simulated model selection. BioData Mining 5(1), 15 (2012)
23. Urbanowicz, R., Kiralis, J., Sinnott-Armstrong, N., Heberling, T., Fisher, J., Moore, J.: Gametes: a fast, direct algorithm for generating pure, strict, epistatic models with random architectures. BioData Mining 5(1), 16 (2012)
24. Velez, D., White, B., Motsinger, A., Bush, W., Ritchie, M., Williams, S., Moore, J.: A balanced accuracy metric for epistasis modeling in imbalanced datasets using multifactor dimensionality reduction. Genet. Epidemiol. 31, 306–315 (2007)

Emergence of Motifs in Model Gene Regulatory Networks

Marcin Zagórski

Institute of Physics, Jagiellonian University, Reymonta 4, 30-059 Kraków, Poland
marcin.zagorski@uj.edu.pl

Abstract. Gene regulatory networks arise in all living cells, allowing the control of gene expression patterns. The study of their circuitry has revealed that certain subgraphs of interactions or motifs appear at anomalously high frequencies. We investigate here whether the overrepresentation of these motifs can be explained by the functional capabilities of these networks. Given a framework for describing regulatory interactions and dynamics, we consider in the space of all regulatory networks those that have a prescribed function. Markov Chain Monte Carlo sampling is then used to determine how these functional networks lead to specific motif statistics in the interaction structure. We conclude that different classes of network motifs are found depending on the functional constraint (multi-stability or oscillatory behaviour) imposed on the system evolution. The discussed computational framework can also be used for predicting regulatory interactions, if only the experimental gene expression pattern is provided.

Keywords: gene regulatory networks, network motifs, transcription factors, cell cycle.

General Description

After billions of years of evolution Earth's life is a very diverse phenomenon, yet all the living organisms are made of simple building blocks called cells. The single cell is a device designed to interpret internal or external signals in order to enhance its survival prospects. One of the key mechanisms responsible for processing available information are regulatory interactions between genes. For instance, when a yeast cell finds itself in the environment rich in sugar it starts to produce enzymes to process this nutrient into energy. If we go down to a molecular level, the sugar presence or absence can be treated as an input signal for a cell's processing unit, *i.e.* gene regulatory network (GRN). The set of interactions between genes along with the gene expression machinery allows all living cells to control their gene expression patterns. In the last decade, our knowledge how any given gene can affect another's expression has been significantly extended through various experiments. For example, small gene networks have been constructed to implement simple functions *in vivo* [3,4], and much larger sets of interactions have been derived from a number of organisms [6,13,7].

L. Vanneschi, W.S. Bush, and M. Giacobini (Eds.): EvoBIO 2013, LNCS 7833, pp. 212–215, 2013.

Therefore it has been possible to show that several subgraphs of interactions ("motifs") arise more frequently than might be expected [14,9,8,16]. In a very recent study [5], the motif statistics were reported for the human regulatory network, indicating overrepresentation of certain structures. Hence, the question of design principles or conditions under which certain motifs appear in biological networks is of great interest.

In [2] by Z. Burda, A. Krzywicki, O.C. Martin and myself published in PNAS 108, 17263-17268 (2011) we propose a computational framework within which gene regulatory networks with a predefined functional capabilities can be sampled *in silico*. Thus, it is possible to study various statistical properties of networks generated with certain constraints imposed. Specifically, the proposed model incorporates microscopic interactions between genes and transcription factors through a weight matrix (genotype). Next, the gene's expression level is determined by deterministic synchronous dynamics with contribution from both excitatory and inhibitory interactions. Having defined transcriptional dynamics and providing initial gene expressions, we can easily obtain gene expression pattern (phenotype) which is interpreted as a function of GRN. By imposing some arbitrary target gene expression pattern (target phenotype) we would like to know which genotypes do lead to this predefined target. In practice, we quantify how well a given genotype is adapted to the imposed pattern by a fitness function depending on a distance between target phenotype and phenotype produced by that genotype.

The main computational difficulty lays in a problem: how from the huge space of all genotypes obtain a sample of genotypes with a high fitness? This goal can be achieved by Markov Chain Monte Carlo (MCMC) method, which generates a biased random walk in the space of genotypes, enforcing at each step the accept/reject Metropolis rule [11]. Note that the MCMC introduces no bias: at large times the *a priori* specified distribution is obtained exactly. Hence, we can understand how phenotypic properties constrain the genotypes, in particular at the level of the architecture of the genetic interactions. Additionally, we can use Metropolis rule to determine which of genetic interactions described by genotype are essential for its functionality. If the element of the genotype matrix corresponding to interaction strength between two genes is set to zero, and the viability of network is lost (the fitness drops and the Metropolis rule rejects modified genotype), the interaction between these genes is considered essential. As a result, a set of all essential interactions that constitutes the gene regulatory network for the underlying genotype is obtained.

A very gratifying point is that obtained GRNs are evolvable and a given target expression pattern can be realized through different topologies. Particularly, we consider two classes of constraints which resemble two types of biological processes: (i) different stable gene expression patterns can be interpreted as different types of cells during cell development, (ii) cyclic gene expression is characteristic for cell cycle, where different genes are excited/inhibited during different stages of cell division process. In order to reveal significant network motifs we compare the number of subgraphs of a given type between generated GRNs and their

randomized versions. The randomization used is that proposed by Maslov and Sneppen [10]: edges are interchanged so that both the in- and out-degrees of network nodes remain unchanged.

In the case of multistability the two node motif with genes being mutually inhibitory and self-activating (double negative feedback loop with autoregulation) is found to be of great importance. Typically in the resulting GRNs, there is one such motif for two fixed points imposed, two for three fixed points imposed and three in case found four steady states. The randomized networks almost always do not have any motifs of this type (see [2] for exact frequencies). Interestingly, this simple network motif is found in various biological gene networks, with a good example being the genetic switch between lysogeny and lysis of the phage λ [12]. Clearly such a pair of genes acts as a bistable between situations with one gene being "on" and the other being "off". When embedded in the whole network this type of motif influence other genes in a downstream effect along the associated tree-like graph structure.

In the case of target phenotypes being periodic in time the bistable switch is not present, and four node motifs like bifan, diamond and "frustrated" loop appear and are highly overrepresented compared to randomized networks (see Fig. 1 for graphical representation). Again, biological gene networks have been found containing some of these motifs [1] the bifan motif being perhaps the most prominent. The function of this motifs treated separately can be understood only for the small network sizes. However, for networks with several genes (as in the discussed study [2]) it is necessary to consider how these motifs cooperate within the overall network, just like parts in a larger machine. More importantly, none of motifs overrepresented for periodic gene expression pattern imposed was found significant in the multiple fixed point scenario, and vice-versa.

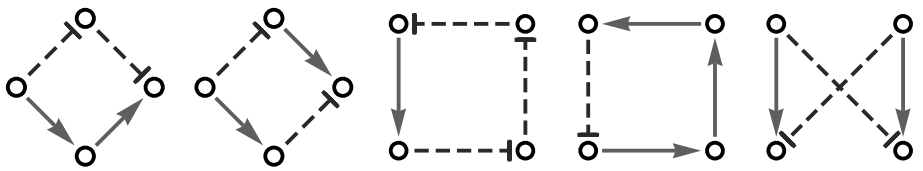


Fig. 1. Network motifs overrepresented in case of time periodic gene expression: incoherent diamonds (from the left: 1st and 2nd), frustrated four-node loops (3rd and 4th), incoherent bifan (5th). The arrows represent activatory (solid) and inhibitory (dashed) interactions.

Hence, we can conclude that different classes of motifs are observed for different types of functional capabilities of GRN. This result is very striking if we realize that no motif structures are incorporated inside the presented framework on any level. Instead motifs emerge from purely random background due to imposed functional patterns and selection pressure. Within the proposed computational framework it is also possible to impose gene expression patterns taken from experimental works (recently we have applied our model to cell cycle profiles of

two yeast species and mammals [15]). Specifically, the question of probability of observing certain interaction between selected genes can be addressed, so the model can be also used as a tool for network structure prediction.

References

1. Alon, U.: Network motifs: theory and experimental approaches. Nature Reviews Genetics 8, 450 (2007)
2. Burda, Z., Krzywicki, A., Martin, O.C., Zagorski, M.: Proc. Natl. Acad. Sci. U.S.A. 108, 17263 (2011)
3. Elowitz, M., Leibler, S.: A synthetic oscillatory network of transcriptional regulators. Nature 403, 335 (2000)
4. Gardner, T., Cantor, C., Collins, J.: Construction of a genetic toggle switch in *Escherichia coli*. Nature 403, 339 (2000)
5. Gerstein, M.B., et al.: Architecture of the human regulatory network derived from ENCODE data. Nature 489, 91 (2012)
6. Herrgard, M., Covert, M., Palsson, B.: Reconstruction of microbial transcriptional regulatory networks. Current Opinion in Biotechnology 15, 70 (2004)
7. Hu, Z., Killion, P., Iyer, V.: Genetic reconstruction of a functional transcriptional regulatory network. Nature Genetics 39, 683 (2007)
8. Lee, T., et al.: Transcriptional regulatory networks in *Saccharomyces cerevisiae*. Science 298, 799 (2002)
9. Ma, H., et al.: An extended transcriptional regulatory network of *Escherichia coli* and analysis of its hierarchical structure and network motifs. Nucleic Acids Research 32, 6643 (2004)
10. Maslov, S., Sneppen, K.: Specificity and stability in topology of protein networks. Science 296, 910 (2002)
11. Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E.: Equation of state calculations by fast computing machines. Journal of Chemical Physics 21, 1087 (1953)
12. Ptashne, M.: A Genetic Switch: Phage λ Revisited. Cold Harbor Spring Laboratory Press, NY (2004)
13. Salgado, H., et al.: Regulondb (version 5.0): *Escherichia coli* k-12 transcriptional regulatory network, operon organization, and growth conditions. Nucleic Acids Research 34, D394 (2006)
14. Shen-Orr, S., Milo, R., Mangan, S., Alon, U.: Network motifs in the transcriptional regulation network of *Escherichia coli*. Nature Genetics 31, 64 (2002)
15. Zagorski, M., Krzywicki, A., Martin, O.C.: Edge usage, motifs and regulatory logic for cell cycling genetic networks. Phys. Rev. E 87, 012727 (2013)
16. Zhu, J., et al.: Integrating large-scale functional genomic data to dissect the complexity of yeast regulatory networks. Nature Genetics 40, 854 (2008)

Author Index