MLSB ^{08}

The second International Workshop on Machine Learning in Systems Biology

13-14 September 2008

Brussels, Belgium

PASCAL2
Pattern Analysis, Statistical Modelling and Computational Learning

Université de Liège GIGA ULg

AIM

FNRS

ECML PKDD 08

BIOMAGNET
Bioinformatics and Modelling: from Genomes to Networks

Belgian Science Policy

FWO VLAANDEREN
Fonds Wetenschappelijk Onderzoek
Research Foundation – Flanders

Editors
Louis Wehenkel
Florence d'Alché-Buc
Yves Moreau
Pierre Geurts

Cover design: Vincent Botta

Contents

MLSB08, the Second International Workshop on Machine Learning in Systems Biology took place in Brussels on September 13-14 2008 in the Palace of the Royal Academy of Belgium.
The aim of this workshop is to contribute to the cross-fertilization between the research in machine learning methods and their applications to complex biological and medical questions by bringing together method developers and experimentalists.

Conference Chairs

Louis Wehenkel and Pierre Geurts, GIGA-Research, University of Liège
Florence d'Alché-Buc, IBISC CNRS FRE 3190, Université d'Evry & URA CNRS 2171, Institut Pasteur, Paris, France
Yves Moreau, ESAT, Katholieke Universiteit Leuven, Belgium

Organizing Support

Michèle Delville, Céline Dizier
Association des Ingénieurs de Montefiore (A.I.M.)

Programme Committee

Florence d'Alché-Buc, University of Evry, France
Christophe Ambroise, University of Evry, France
Pierre Geurts, University of Liège, Belgium
Mark Girolami, University of Glasgow, UK
Samuel Kaski, University of Helsinki, Finland
Kathleen Marchal, Katholieke Universiteit Leuven, Belgium
Elena Marchiori, Vrije Universiteit Amsterdam, The Netherlands
Yves Moreau, Katholieke Universiteit Leuven, Belgium
Gunnar Rätsch, FML, Max Planck Society, Tübingen
Juho Rousu, University of Helsinki, Finland
Céline Rouveirol, University of Paris XIII, France
Yvan Saeys, University of Gent, Belgium
Rodolphe Sepulchre, University of Liège, Belgium
Koji Tsuda, Max Planck Institute, Tuebingen
Jacques Van Helden, Université Libre de Bruxelles, Belgium
Kristel Van Steen, University of Liège, Belgium
Jean-Philippe Vert, Ecole des Mines, France
Louis Wehenkel, University of Liège, Belgium
David Wild, University of Warwick, UK
Jean-Daniel Zucker, University of Paris XIII, France

Sponsors

The organizers gratefully acknowledge the following sponsors who have provided financial support or/and precious help for the organization of the workshop.

Association des Ingénieurs de Montefiore, ULg

Royal Academy of Belgium

GIGA research, University of Liège

PASCAL 2 European network of excellence

Fonds National de la Recherche Scientifique

Fonds Wetenschappelijk Onderzoek, Vlaanderen

Biomagnet Iap P6/25

ECML/PKDD 2008

Schedule

Saturday, September 13th

9h30-9h45 Welcome

9h45-10h45 Invited Talk

Pamela Silver (Harvard Medical School), *Designing Biological Systems* (*p10*)

10h45-11h15 Coffee break

11h15-12h30 Session 1

11h15-11h40 Saso Dzeroski and Ljupco Todorovski. *Equation Discovery for Systems Biology* (*p17*)

11h40-12h05 Karoline Faust, Jérôme Callut, Pierre Dupont, and Jacques van Helden. *Metabolic Pathway Inference using Random Walks and Shortest-Paths Algorithms* (*p27*)

12h05-12h30 Alexandre Irrthum and Louis Wehenkel. *Predicting gene essentiality from expression patterns in Escherichia coli* (*p39*)

12h30-12h55 Invited Talk

Alain Chariot (ULg). *Deciphering the molecular mechanisms underlying human diseases through interactome studies: a molecular approach* (*p12*)

12h55-15h00 Lunch break and poster session

15h00-16h00 Invited Talk

Lukas Käll (University of Washington). *Semi-supervised machine learning for shotgun proteomics* (*p10*)

16h00-16h30 Coffee break and poster session

16h30-17h20 Session 2

16h30-16h55 Artem Sokolov and Asa Ben-Hur. *A Structured-Outputs Method for Prediction of Protein Function* (*p49*)

16h55-17h20 Omer Sinan Sarac, Rengul Cetin-Atalay, and Volkan Atalay. *GOPred: Combining classifiers on the GO* (*p59*)

17h20-17h45 Invited Talk

Heribert Hirt (URGV Plant Genomics Institute & University of Vienna). *Phosphoproteomic approaches to study stress signal transduction networks in plants* (*p12*)

19h00 Conference dinner at the restaurant "L'atelier"

Sunday, September 14th

9h00-10h00 Invited Talk

Yoav Freund (UCSD). *From microscopy images to models of cellular processes* (*p*11)

10h00-10h30 Coffee break

10h30-11h45 Session 3

10h30-10h55 Koenraad Van Leemput and Alain Verschoren. *Modeling Networks as Probabilistic Sequences of Frequent Subgraphs* (*p*67)

10h55-11h20 Michalis Titsias, Neil Lawrence, and Magnus Rattray. *Sampling for Gaussian Process Inference* (*p*77)

11h20-11h45 Selpi, Christopher H. Bryant, and Graham Kemp. *Using mRNA Secondary Structure Predictions Improves Recognition of Known Yeast Functional uORFs* (*p*85)

11h45-12h10 Invited Talk

Marc Muller (ULg). *The zebrafish as a small vertebrate model system for bone development and homeostasis* (*p*13)

12h10-14h15 Lunch break and poster session

14h15-15h15 Invited Talk

Lodewyk Wessels (Netherlands Cancer Institute). *Outcome prediction in breast cancer* (*p*11)

15h15-15h45 Coffee break and poster session

15h45-16h35 Session 4

15h45-16h10 Fan Shi, Geoff Macintyre, Christopher Andrew Leckie, Izhak Haviv, Alex Boussioutas, and Adam Kowalczyk. *A Bi-ordering Approach to Linking Gene Expressions with Clinical Annotations in Cancer* (*p*95)

16h10-16h35 Vincent Botta, Sarah Hansoul, Pierre Geurts, and Louis Wehenkel. *Raw genotypes vs haplotype blocks for genome wide association studies by random forests* (*p*105)

16h35-17h00 Invited Talk

Bernard Thienpont (KU Leuven). *Endeavour pinpoints genes causing cardiac defects in regions identified by aCGH* (*p*14)

17h00-17h15 Closing

Invited talks

Designing Biological Systems

Pamela A. Silver

Professor, Department of Systems Biology, Harvard Medical School and Director of the Harvard University Graduate Program in Systems Biology

Abstract

Biology presents us with an array of design principles that extend beyond what is normally found in silico. However, we don't yet know how to make facile use what we know and there is a lot more to learn. As a start, we are interested in using the foundations of biology to engineer cells in a simple and logical way to perform certain functions. In doing so, we learn more about the fundamentals of biological design as well as engineer useful devices with myriad applications. For example, we are interested in building cells that can perform specific tasks, such as counting, measuring and remembering past events. Moreover, we design and construct proteins and cells with predictable biological properties that not only teach us about biology but also serve as potential therapeutics, cell-based sensors and factories for generating bio-energy.

Semi-supervised machine learning for shotgun proteomics

Lukas Käll

Department of Genome Sciences, University of Washington

Abstract

Shotgun proteomics refers to the analysis of protein mixtures by cleaving the proteins with an enzyme, detecting the resulting peptides with tandem mass spectrometry and subsequently identifying the peptides with database search algorithms. The approach is currently considered the most accurate way to determine the protein content of a complex biological mixture. A limitation of existing machine learning efforts to improve peptide identification in shotgun proteomics datasets are that they are based on fixed training sets and are hence unable to compensate easily for variations in mass spectrometry conditions. Instead of curating representative training sets for individual conditions, which in most cases is not practically feasible, we have devised algorithms that are capable of learning directly from the individual shotgun proteomics datasets that we want to classify. Using semi-supervised learning to discriminate between correct and incorrect spectrum identifications we correctly assign peptides to up to 77% more spectra, relative to a fully supervised approach.

From microscopy images to models of cellular processes

Yoav Freund

Professor, Computer Science and Engineering, UCSD

Abstract

The advance of fluorescent tagging and of confocal microscopy is allowing biologists to image biochemical processes at a level of detail that was unimaginable just a few years ago. However, as the analysis of these images is done mostly by hand, there is a severe bottleneck in transforming these images into useful quantitative data that can be used to evaluate mathematical models.

One of the inherent challenges involved in automating this transformation is that image data is highly variable. This requires a recalibration of the image processing algorithms for each experiment. We use machine learning methods to enable the experimentalist to calibrate the image processing methods without having any knowledge of how these methods work. This, we believe, will allow the rapid integration of computer vision methods with confocal microscopy and open the way to the development of quantitative spatial models of cellular processes.

For more information, see
`http://seed.ucsd.edu/~yfreund/NewHomePage/Applications/Biomedical_Imaging.html`.

Outcome prediction in breast cancer

Lodewyk Wessels

Professor, Bioinformatics and Statistics group, Netherlands Cancer Institute, Amsterdam, The Netherlands

Abstract

Background: Michiels et al. (Lancet 2005; 365: 488-92) employed a resampling strategy to show that the genes identified as predictors of prognosis from resamplings of a single gene expression dataset are highly variable. The genes most frequently identified in the separate resamplings were put forward as gold . On a higher level, breast cancer datasets collected by different institutions can be considered as resamplings from the underlying breast cancer population. The limited overlap between published prognostic signatures confirms the trend of signature instability identified by the resampling strategy. Six breast cancer datasets, totaling 947 samples, all measured on the Affymetrix platform, are currently available. This provides a unique opportunity to employ a substantial dataset to investigate the effects of pooling datasets on classifier accuracy, signature stability and enrichment of functional categories.

Results: We show that the resampling strategy produces a suboptimal ranking of genes, which can not be considered to be gold . When pooling breast cancer datasets, we observed a synergetic effect on the classification performance in 73% of the cases. We also observe a significant positive correlation between the number of datasets that is pooled, the validation performance, the number of genes selected, and the enrichment of specific functional categories. In addition, we have tested five hypotheses that have been postulated as an explanation for the limited overlap of signatures.

Conclusions: The limited overlap of current signature genes can be attributed to small sample size. Pooling datasets results in more accurate classification and a convergence of signature genes. We therefore advocate the analysis of new data within the context of a compendium, rather than analysis in isolation.

Deciphering the molecular mechanisms underlying human diseases through interactome studies: a molecular approach

Alain Chariot

Laboratory of Medical Chemistry, Unit of Signal Transduction, GIGA-research, University of Liège, Belgium

Abstract

Establishing the interactome of any given signalling protein is a powerful approach in order to better understand what its biological roles are but also to precise to which extent this interactome is specifically altered in human diseases. We have been using the yeast-two-hybrid approach in order to decipher the signalling pathways regulated by two families of transcription factors, namely NF-κ and IRFs. Both families have deregulated, constitutive activities in a variety of solid and haematological cancers as well as in chronic inflammatory and neurodegenerative disorders.

Our recent interactome data not only highlighted where, when and how these signalling proteins are involved in signal transduction but also helped us to better understand how the post-translational modifications of those proteins regulate their function. We will present examples of ongoing research projects in our laboratory dedicated to the establishment of interacting networks and demonstrate how those networks help to better understand why their deregulations lead to diseases.

Phosphoproteomic approaches to study stress signal transduction networks in plants

Heribert Hirt

URGV Plant Genomics Institute, Paris, France & Department of Plant Molecular Biology, University of Vienna, Austria

Abstract

We are interested to study protein kinase networks that function in environmental stress responses. As such we have identified the MEKK1-MKK2-MPK4 signalling pathway which plays a role in resistance to both biotic and abiotic stresses (Teige et al., 2004, Nakagami et al., 2006, Brader et al., 2007). To obtain a more global view on signalling, the state of multiple signal pathways under any one condition and time is monitored by phosphoproteomics and phosphosite-specific microarrays (de la Fuente van Bentem et al., 2007). On the basis of these data, system hypotheses are developed to undergo reiterative experimental testing and remodeling. As an exemple for the usefulness of this approach, I will discuss recent work on the plant-microbe interaction system of Agrobacterium and Arabidopsis.

1. Teige, M., Scheikl, E., Eulgem, T., Doczi, R., Ichimura, K., Shinozaki, K., Dangl, J.L., and Hirt, H. (2004) The MKK2 pathway mediates cold and salt stress signaling in Arabdiopsis. *Mol. Cell* 15, 141-152.
2. Nakagami, H., Soukupova, H., Schikora, A., Zarsky, V. and Hirt, H. (2006) A mitogen-activated protein kinase kinase kinase mediates reactive oxygen species homeostasis in Arabidopsis. *J. Biol. Chem.* 28, 3267-78.
3. Brader, G., Djamei, A., Teige, M., Palva, T. Hirt, H. (2007) The MAP kinase kinase MKK2 affects diseasse resistance in Arabidopsis. *Mol. Plant Micr. Int.* 20, 589-596.
4. van Bentem, S. and Hirt, H (2007) Using phosphoproteomics to reveal signalling dynamics in plants. *Trends Plant Sci.* 12, 404-409
5. Djamei, A., Pitzschke, A., Nakagami, H., Rajh, I., Hirt, H. (2007) Trojan horse strategy in Agrobacterium transformation: Abusing MAPK defense signaling. *Science* 318, 453 – 456.

The zebrafish as a small vertebrate model system for bone development and homeostasis

Marc Muller[1], Jessica Aceto[1], Julia Dalcq[1], Peter Alestrom[2], Rasoul Nourizadeh-Lillabadi[2], Roland Goerlich[3], Viktoria Schiller[3], Christoph Winkler[4], Jörg Renn[4], Matthias Eberius[5], Klaus Slenzka[6]

[1]Laboratoire de Biologie Moléculaire et de Génie Génétique, Université de Liège

[2]Dept of Basic Sciences and Aquatic Medicine, Norwegian School Veterinary Science, Oslo, Norway

[3]Dept. of Molecular Biotechnology, RWTH Aachen

[4]Department of Biological Sciences, National University of Singapore

[5]LemnaTec GmbH, Würselen

[6]Orbitale Hochtechnologie Bremen (OHB)-System AG, Bremen

Abstract

Small fish models, mainly zebrafish (Danio rerio) and medaka (Oryzias latipes), have been used for many years as powerful model systems for vertebrate developmental biology. Moreover, these species are increasingly recognized as valuable systems to study vertebrate physiology, pathology, pharmacology and toxicology. In recent years, analysis of gene function by mutation or genetic manipulation has shown that the homologs of many genes previously described to be involved in bone development and homeostasis in mammals also play very similar roles in small fish species. Bone physiology is affected by homologous genes in mammals and zebrafish. Thus, small fish models represent a valuable tool to investigate bone development and pathology.

Small fish species present many advantages for studying development, such as transparency of the embryos, external development, possibility for large scale mutagenesis screening, rapid development. These include large number of embryos from one single clutch, small size, easy containment in water tanks. Many technologies for visualizing and characterizing bones, such as specific staining or fluorescent transgenic animals, have been adapted to small fish species and can be routinely performed on large numbers of larvae. Furthermore, its genome sequencing and annotation is close to completion making whole genome analysis feasible.

Our principal objective is to study bone pathologies in zebrafish, such as osteoporosis induced by menopause or prolonged space flight. We investigate the changes induced by mutations, bone-metabolizing drugs or microgravity in small fish species. One type of approach is to combine whole genome approaches, such as microarray expression analysis, chromatin immunoprecipitation (ChIP) or proteomics with a special emphasis on bone-related genes. Data are obtained by microgravity simulation on ground and compared to the changes observed in space. A complementary strategy is to carry out automated in vivo real time observations of transgenic larvae expressing a fluorescent reporter protein in bone-related structures.

Endeavour pinpoints genes causing cardiac defects in regions identified by aCGH

Bernard Thienpont[1], **R. Barriot**[3], **P. Van Loo**[3], **L. Tranchevent**[3], **M. Gewillig**[2], **Y. Moreau**[3], **K. Devriendt**[1]

[1]Center for Human Genetics, University of Leuven, Belgium
[2]Paediatric Cardiology Unit, University of Leuven, Belgium
[3]Department of Electrical Engineering, ESAT-SCD, University of Leuven, Belgium

Abstract

Array Comparative Genomic Hybridisation (aCGH) is a novel tool for high-resolution detection of submicroscopic chromosomal insertions or deletions (indels). It opens opportunities in diagnostics as well as in the identification of novel loci involved in the patients phenotype. We analysed 130 patients with an idiopathic syndromic congenital heart defect (CHD) by array-CGH at 1Mb resolution, resulting in the detection of causal imbalances in 22 patients (17%).

All indels as well as indels and gene mutations described in the CHD literature were collected in a centralized repository CHDWiki, that allows a collaborative annotation of the genome. In 50% of the cases (11/22) the indel affects a gene annotated to cause CHDs.

The other indels pinpoint regions that contain novel candidate genes for CHD. To identify these genes, an /in silico/ prioritisation algorithm (based on Endeavour) was developed. Extensive /in silico/ testing demonstrated a high discriminative power. The results of prioritizing genes from the indel regions were further verified by analysing the expression of 45 high ranking genes by /in situ/ hybridisation on developing zebrafish embryos. These analyses supported the involvement of two novel genes in human CHD: /BMP4/and /HAND2/.

In conclusion, we show that aCGH can provide an etiological diagnosis in 17% of patients with a syndromic CHD. It can moreover contribute to the discovery of genes causing CHD in humans, and drive research on how they contribute to normal and pathogenic cardiovascular development.

Contributed papers

List of papers

Equation Discovery for Systems Biology

Sašo Džeroski and Ljupčo Todorovski

[1] Jozef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
[2] Faculty of Administration, University of Ljubljana
Gosarjeva 5, SI-1000 Ljubljana, Slovenia

Abstract. Reconstructing biological networks is at the heart of systems biology. While many approaches exist for reconstructing network structure, few approaches reconstruct the full dynamic behavior of a network. We survey such approaches that originate from computational scientific discovery, a subfield of machine learning. These take as input time course data, as well as domain knowledge, such as partial knowledge of the network structure, and output differential equation models describing both the structire and dynamics of the networ. We demonstrate the use of these approaches on illustrative example tasks of reconstructing the complete dynamics of biochemical reaction networks.

1 Introduction

The (re)construction of biological networks, including metabolic, regulatory and signalling networks, is of fundamental and immediate importance to the emerging field of computational systems biology [1]. The task to be addressed first in this context is the reconstruction of the structure of the network: The stoichiometric matrix identifies the reactions in the network with the participating molecular compounds (rows represent compounds, columns reactions). For metabolism, the networks often focus on just the metabolites, while for regulatory and signalling networks the inclusion of proteins is essential.

The dynamic behavior of biological networks is typically modelled by ordinary differential equation (ODE) models. Besides the dependences between compounds in the network, as specified by network structure, ODE models specify the exact nature of these dependencies through the functional form of the ODEs and their constant parameters (e.g., reaction rates). In a typical approach to ODE modeling of a biological network, a human domain expert specifies the structure of the network and the functional form of the ODEs. Time course data about the behavior of the target reaction network can be used to determine the values of the constant parameters in the ODEs.

Determining a set of ODEs from given time course data is referred to as system identification. The task of determining the functional form of a set of ODEs is referred to as structure identification. The task of determining appropriate values for the constant parameters is called parameter estimation. In this article, we will discuss approaches to performing both of these tasks simultaneously. The approaches we survey come from the area of machine learning [2, 3], more specifically computational scientific discovery [4, 5], and are directly relevant to but not widely known in the systems biology community.

2 Computational scientific discovery of ODE models

Computational scientific discovery (CSD) [4, 5] is concerned with developing computer programs that automate or support some aspects of scientific discovery. The earliest and most prominent CSD systems, such as BACON [4], dealt with the problem of equation discovery, finding scientific laws in the form of equations. While early CSD considered algebraic equations, they were later extended to learn ODE models from time course data [6].

CSD is a subfield of machine learning [2, 3] and artificial intelligence [7]. From this broader context, it inherits the fundamental approach of problem solving as heuristic search. In particular, CSD programs for ODE discovery would search the space of ODE structures (functional forms) guided by heuristics related to the degree of fit of the ODEs to the data. To evaluate these, parameter estimation needs to be performed for each ODE structure: as we are often interested in structures nonlinear in the parameters, computationally expensive nonlinear optimization has to be used.

Another source of computational complexity is the size of the space of possible ODE model structures: This is typically huge and can easily be infinite. Of crucial importance is thus to define the space of ODE structures so as to keep it small and pertinent to capturing the dynamics of the modelled system. To achieve this, the use of domain knowledge in equation discovery has been proposed [8].

Different types of domain knowledge can be used in ODE discovery. We can start from existing ODE models for the system at hand (that are partial/incomplete/inaccurate) and revise/improve them in light of observed time course data. We can also provide a set of basic components as building blocks from which ODE models can be built (akin to compositional modelling [9]). Finally, we can provide a set of constraints that the ODE models we are willing to consider have to satisfy. Common to all of these is the explicit (declarative) statement of the modelling assumptions made concerning the space of ODE models considered. Below we briefly describe several CSD approaches to ODE discovery that can use domain knowledge of these types and illustrate them with examples related to biological networks.

3 Constrained induction of polynomial equations

The CSD system CIPER (Constrained induction of polynomial equations for regression) [10, 11] considers the space of polynomial equations, which are linear in the parameters, and uses linear regression for parameter estimation. It performs heuristic search of this space, ordered by the relation of subpolynomial on structures: a polynomial structure is a subpolynomial of another, if we can obtain the first from the second by omitting some parts (terms or appearances of variables in terms). The search proceeds from simple structures (starting with a constant term only) to more complex ones by adding new linear terms or multiplying existing terms with a variable.

Fig. 1. A reaction network (a), consisting of six reactions, that was successfully reconstructed from simulated data and a partial specification of the network structure by constrained induction for polynomial regression (CIPER) [11]. The parts given in bold are assumed not to be known for the reconstruction task. The partial specification of the equation structure (b) is derived from the known part of the network: the polynomials in the partial structure have to be subpolynomials of the corresponding polynomials found by CIPER and are supplied to CIPER as subsumption constraints. Simulated data were obtained from the complete equations (c), which were successfuly reconstructed when CIPER was given both simulated data and a partial structure. Given only simulated data, CIPER searched a much larger space of structures and failed to reconstruct correctly the most complex equations, i.e., the ones for $\dot{x_1}$ and $\dot{x_2}$.

(a) A partially specified network of reactions

$\{x_5, \mathbf{x_7}\} \rightarrow \{x_1\}$; $\{x_1\} \rightarrow \{x_2, x_3\}$
$\{x_1, x_2, \mathbf{x_7}\} \rightarrow \{x_3\}$; $\{x_3\} \rightarrow \{x_4\}$
$\{x_4\} \rightarrow \{x_2, \mathbf{x_6}\}$; $\{\mathbf{x_4}, \mathbf{x_6}\} \rightarrow \{\mathbf{x_2}\}$

(b)Partial structure/(c)Full equations	$\dot{x_1} = 0.8 \cdot x_5 \cdot x_7 - 0.5 \cdot x_1 - 0.7 \cdot x_1 \cdot x_2 \cdot x_7$
	$\dot{x_2} = 0.7 \cdot x_1 + 0.2 \cdot x_4 + 0.1 \cdot x_4 \cdot x_6 - 0.3 \cdot x_1 \cdot x_2 \cdot x_7$
$\dot{x_1} = -c \cdot x_1 + c \cdot x_5 - c \cdot x_1 \cdot x_2$	$\dot{x_3} = 0.4 \cdot x_1 + 0.3 \cdot x_1 \cdot x_2 \cdot x_7 - 0.2 \cdot x_3$
$\dot{x_2} = c \cdot x_1 + c \cdot x_4 - c \cdot x_1 \cdot x_2$	$\dot{x_4} = 0.5 \cdot x_3 - 0.7 \cdot x_4 \cdot x_6$
$\dot{x_3} = c \cdot x_1 + c \cdot x_1 \cdot x_2 - c \cdot x_3$	$\dot{x_5} = -0.6 \cdot x_5 \cdot x_7$
$\dot{x_4} = c \cdot x_3 - c \cdot x_4$	$\dot{x_6} = 0.2 \cdot x_4 - 0.8 \cdot x_4 \cdot x_6$
$\dot{x_5} = -c \cdot x_5$	$\dot{x_7} = -0.1 \cdot x_1 \cdot x_2 \cdot x_7 - 0.1 \cdot x_5 \cdot x_7$

For its search, the original CIPER uses a heuristic that combines model error and model complexity in an ad-hoc fashion. The latest version of CIPER [12] uses the minimum-description length principle (MDL) [13] to combine these in a principled manner. It can take into account subsumption constraints, specifying a structure that should subsume the model to be found: These can be used, for example, to specify partially known equation structures. CIPER [14] can also find equations for several variables simultaneously: This is beneficial for modeling reaction networks, as variables that appear together in a reaction typically share terms in the corresponding equations. Note that CIPER has been designed primarily for algebraic equations: It handles ODEs by numerically introducing time derivatives of the system variables.

Constraints in CIPER are useful in the context of reconstructing reaction networks [11] from partial structures (see Figure 1). If we consider a simplified version of the S-system [15], where simple products are used instead of products of powers, we obtain polynomial ODEs. Given time course data obtained by simulating the ODEs and the constraints resulting from the partial network (i.e., polynomial) structure, CIPER successfully reconstructs the ODE model. Without the constraints, however, the reconstruction is not completely successful: This illustrates the crucial role that domain knowledge can play.

4 Grammar-based equation discovery

To represent the possible space of ODE structures, we can view it as a language, with individual equation structures being sentences. A formal grammar can then be used to describe the language. The equation discovery system LAGRAMGE [16] uses the formalism of context-free grammars (CFG) for this purpose.

Formally, a CFG is a four-tuple (S,N,T,P), where S is a starting symbol from N, the set of non-terminals, T is a set of terminals, and P is a set of productions. Nonterminals (or syntactic categories), represent classes of subexpressions or phrases in the language represented by the grammar (e.g., a nounphrase in English; a term in a polynomial). Terminals are the symbols that actually appear in the sentences of the language (e.g., words and punctuation in English; variable names or arithmetical operators). Productions (or rewrite rules) specify how a nonterminal can be replaced by a sequence of (terminals and) nonterminals (e.g., a determiner followed by a noun is a nounphrase; a term is a polynomial; adding a term to a polynomial we obtain a polynomial).

Given a context free grammar, we can check whether a sentence/expression belongs to the language defined by the grammar (parse task) or generate expressions that belong to the language (generate task). For both purposes, we use the notion of a parse tree, which describes the way a certain expression can be derived using the grammar productions.

LAGRAMGE performs heuristic (or exhaustive) search over the space of ODE model structures defined by depth-bounded derivation trees for a given CFG. In particular, beam search is used, which keeps several alternative ODE structures found to be best so far. The heuristic (quality criterion) used is a combination of the ODE model error and its complexity. To calculate the ODE model error with respect to given time course data, LAGRAMGE fits the constant parameters by non-linear optimization using the ALG 717 [19] generalization of the NL2SOL adaptive nonlinear least-squares algorithm [20].

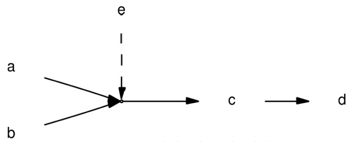
Grammars are an expressive formalism for representing many different types of domain knowledge, including existing models to be revised, incomplete/partial models, and knowledge-based building blocks for modelling in a particular domain [8]. Note, however, that a grammar is specific to a given modelling task at hand. Also, grammar formalisms make little contact with the formalisms used by mathematical modellers and scientists and are thus difficult to use.

5 Representing process-based models

The representation of process-based models (PBMs) and process-based domain knowledge (PBDK) is more general and accessible to scientists and engineers, who often state their explanations in terms of processes that govern the behaviour of an observed dynamic system. It also connects the explanatory and predictive aspects of modelling, by directly linking processes to the mathematical formulations cast in terms of equations. A basic set of generic processes or process classes can be identified for a domain of interest: together with some formulations cast in terms of equations, this constitutes domain knowledge than can

Fig. 2. Process-based models and domain knowledge. (a) An example network of two reactions and its corresponding ODE model according to the S-system formalism. (b) Process-based domain knowledge (PBDK) for ODE models based on the S-system formalism ready to use by LAGRAMGE2.0. There is only one class of processes, corresponding to reactions, where each reaction gives rise to a product of powers of the concentrations of participating substrates, products and enzymes: The combining scheme adds up the influences of all reactions in which a component participates to obtain its rate of change. (c) A metabolic system (a) taken from Arkin and Ross [18] and used by Gennemark and Wedelin [17] to evaluate their system on the task of reconstructing the dynamics of the system from simulated data. The corresponding ODEs are given in the Appendix of [17], equations (3)-(12). (d) LAGRAMGE 2.0 PBDK for modelling the metabolic system from (c) and the three reactions in the corresponding PBM.

a).



Reaction({e},{a,b},{c})
Reaction({},{c},{d})

$$\dot{a} = -\beta_a \cdot e \cdot a^{\gamma_a} \cdot b^{\gamma_b}$$
$$\dot{b} = -\beta_b \cdot e \cdot a^{\gamma_a} \cdot b^{\gamma_b}$$
$$\dot{c} = \alpha_c \cdot e \cdot a^{\gamma_a} \cdot b^{\gamma_b} - \beta_c \cdot c^{\gamma_c}$$
$$\dot{d} = \alpha_d \cdot c^{\gamma_c},$$

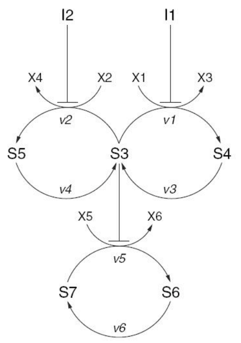
b).

```
type Concentration is nonnegative_real_number

type Substance is Concentration
type Substances is set(Substance)

type Enzyme is Concentration
type Enzymes is set(Enzyme)

process class Reaction(Enzymes es, Substances ins,
     Substances prods)
condition ins ∩ prods = ∅
expression ∏e∈es e * ∏i∈ins pow(i, const(_, 0, 1, Inf))

combining scheme Biochemical_kinetics(Substance s)
ṡ = ∑s∈prods const(_, 0, 1, Inf) ∗ Reaction(es, ins, prods))
- ∑s∈ins const(_, 0, 1, Inf) ∗ Reaction(es, ins, prods))
```

d).

```
process class Reaction(Concentration E, Concentration I,
expression const(vr, 0, 1, 50) * saturation(O)
 - const(vf, 0, 1, 50) * saturation(I)*inhibition(E)

function class Saturation(Concentration c)

function class NoSaturation is Saturation
     expression c

function class MichaelisMenten is Saturation
     expression c / (c + const(kd, 0.1, 1, 50))

function class Inhibition(Concentration e)
     expression 1/(1+e/const(ki, 0.1, 1, 50))

combining scheme Biochemical_Kinetics(Concentration c)
ċ = ∑Reaction(e, s, o) − ∑Reaction(e, i, o)

Reaction(I2,S3,S5), Reaction(I1,S3,S4), Reaction(S3,S6,S7)
```

c.)



be re-used across different modelling tasks in the same domain. In the domain of population dynamics, processes include the growth and decay of a population or interactions between species [21], while in system biology processes correspond to biochemical reactions.

Several formalisms for representing PBMs and PBDK have been proposed recently. Todorovski and Džeroski [22–24, 8] propose a formalism for PBDK that comprises three components: a hierarchy of variable types, a hierarchy of process and function classes, and a combining scheme. The processes and functions relate variable types and specify model structures for individual processes, while the combining scheme specifies how the models of individual processes are combined into a model of the entire observed system. Figure 2b depicts PBDK for modeling biochemical reactions (in the S-system style [15]) expressed in this formalism and an example reaction network, its processes (2a) and corresponding equation structures. Figure 2d depicts PBDK for the metabolic system from Figure 2c.

Fig. 3. Process-based domain knowledge for modeling metabolic networks (a) and a process-based model (b) of the network depicted in Figure 2c. The PBM consists of six processes arising from the three instances of the generic process reaction needed to model the system, as described in the PBDK given in Figure 3a.

a).

```
generic process Reaction
variables E{concentration}, I{concentration}, O{concentration}
processes Positive_Flux(I, O), Negative_Flux(E, I, O)

generic process Positive_Flux_No_Saturation{Positive_Flux}
variables I{concentration}, O{concentration}
parameters vr[0:50]
equations D[I,t,1] = vr * O, D[O,t,1] = -D[I,t,1]

generic process Positive_Flux_Saturated{Positive_Flux}
variables I{concentration}, O{concentration}
parameters vr[0:50], kd[0.1:50]
equations D[I,t,1] = vr * O / (O + kd), D[O,t,1] = -D[I,t,1]

generic process Negative_Flux_No_Saturation_Inhibited{Negative_Flux}
variables E{concentration}, I{concentration}, O{concentration}
parameters vf[0:50], ki[0.1:50]
equations D[O,t,1] = vf * I / (1 + E/ki), d[I,t,1] = -d[O,t,1]

generic process Negative_Flux_Saturated_Inhibited{Negative_Flux}
variables E{concentration}, I{concentration}, O{concentration}
parameters vf[0:50], kd[0.1:50], ki[0.1:50]
equations D[O,t,1] = vf * I / ((I + kd) * (1 + E/ki)), d[I,t,1] = -d[O,t,1]
```

b).

```
process Reaction(I2, S3, S5)
process Positive_Flux_Saturated(S3, S5, vr = 1, kd = 5)
process Negative_Flux_Saturated_Inhibited(I2, S3, S5, vf = 5, kd = 5, ki = 1)

process Reaction(I1, S3, S4)
process Positive_Flux_Saturated(S3, S5, vr = 1, kd = 5)
process Negative_Flux_Saturated_Inhibited(I1, S3, S4, vf = 5, kd = 5, ki = 1)

process Reaction(S3, S7, S6)
process Positive_Flux_Saturated(S7, S6, vr = 1, kd = 5)
process Negative_Flux_Saturated_Inhibited(S3, S7, S6, vf = 10, kd = 5, ki = 1)
```

While the above formalism can be used to represent PBDK, Langley et al. [25, 26] propose a formalism for representing both PBDK and PBMs. This formalism uses generic processes to describe PBDK and specific processes (with specific variables and constant parameter values) to describe PBMs. Figure 3b depicts a process-based model of the metabolic system from Figure 2c. The PBDK in Figure 3a is a special case of domain knowledge/generic processes describing irreversible and reversible chemical reactions as well as inhibition and activation. Such general domain knowledge for modelling metabolic kinetics has been given by Langley et al. [27].

6 Learning process-based models

LAGRAMGE2.0 [22–24, 8] induces PBMs by transforming PBDK into grammars and applying LAGRAMGE in turn. LAGRAMGE2.0 expects as input PBDK as described above (variable types, processes and functions, combining schemes), as well as a modelling task specification, which lists the measured variables and their types and the classes of processes that are expected to appear in the model. Given these, LAGRAMGE2.0 matches the type of variable from the task specification against the types of variables in the process and function classes and transforms the latter into grammar productions specifying modelling alternatives for individual processes. Similarly, the combining scheme is transformed into a grammar production that puts process models together in a single model of the entire system. The obtained grammar specifies the space of ODE models that LAGRAMGE has to search to find a model that optimally fits the observed system behaviour (time course data).

IPM [26], on the other hand, performs heuristic search directly through the space of PBMs. Given a modelling task specification, IPM instantiates generic processes into specific ones that represent model components. Again, IPM searches through the space of combinations of model components in order to find the optimal one. For each candidate model, IPM performs full simulation of the model equations and matches the simulated against the observed behaviour. It is thus capable of inducing models that include unobserved system variables, i.e., variables whose values have not been directly measured/observed. IPM has been applied in the domain of biochemical kinetics as reported in [27], addressing the task of modelling glycolysis from measured data [28].

Note that the IPM search through the space of all combinations of model components leads to a search space whose size grows exponentially with the number of processes included in the model. To make this strategy feasible for complex domains, one must add structural constraints, specifying, e.g., which processes should be included in the system or which processes are mutually exclusive. The HIPM system [29] accepts structural constraints stated as a hierarchy of generic processes.

7 Other recent work

Recent work in CSD related to the discovery of biochemical reaction networks includes work on learning qualitative models of metabolic [30] and genetic [31] networks. Garet et al. [30] learn qualitative differential equations, which have the same functional form as ODE models for the S-system formalism, with products of variables (instead of powers thereof), but no specific values for the constant coefficients. Zupan et al. [31] reconstruct qualitative genetic networks from the outcomes of knock-out and overexpression experiments and background knowledge (kown gene-to-gene and gene-to-outcome interactions).

The above methods infer the structure of a network, without describing its dynamic behavior. Many methods address this task, a survey of which is given by Price and Schmulevich [1]. Network structure can be reconstructed by using information from the literature and databases, or by reverse engineering from genome-wide data on transcriptomics and proteomics [32]. In the latter case, steady state data [33] or time course data [34] can be used as input. An example is the method by Arkin and Ross [18], where a factor analysis of the correlations between time course data on measured variables is conducted and the results are manually interpreted to arrive at a network structure.

A few approaches have explicitly addressed the task of reconstructing both network structure and dynamics, two of which come from the area of evolutionary computation. Koza et al. [35] use genetic programming to reconstruct a metabolic network, where simulated data and information on the types of reactions are taken into account. Kikuchi et al. [36] use a genetic algorithm to reconstruct a genetic network defined in the S-system formalism [15]. Finally, a recent approach by Gennemark and Wedelin [17] performs heuristic search over an ad-hoc defined space of ODE structures to rediscover a metabolic [18] and a genetic network [36].

8 Outlook

The task of reconstructing both the structure and dynamics of biochemical reaction networks is of central interest to computational systems biology. In this article, we have given a survey of methods that perform this task, taking as input time course data, as well as different types of domain kowledge (such as partial network structure). Given that the task is data intensive, the ability of these systems to leverage the data with domain knowledge (and potentially reduce the amount of data needed) is a key feature of interest.

Many challenges remain to be addressed for the successful use of such methods. One of these is the task of parameter identification for ODE structures from short time courses, which are the norm in systems biology. Another task is the casting of domain knowledge for different formalisms that are frequently used to model reaction networks (such as the S-system) into a form usable by computational discovery approaches. Finally, while the approaches outlined above could in principle use the output of network structure reconstruction approaches, it still remains an open issue how to formulate discrete network structures as domain knowledge for discovering models of the dynamic behavior of networks.

Acknowledgements. An extended and revised version of this overview article has been accepted for publication and will appear in *Current Opinion in Biotechnology*, 19:360–368 under the title *Equation discovery for systems biology: finding the structure and dynamics of biological networks from time course data.*

We acknowledge the support of the EU funded project IQ (Inductive Queries for Mining Patterns and Models).

References

1. Price ND, Shmulevich I: **Biochemical and statistical network models for systems biology**. *Curr Opin Biotechnol* 2007, **18**:365-370.
2. Langley P: **Elements of Machine Learning**. San Francisco: Morgan Kaufmann; 1996.
3. Mitchell TM: **Machine Learning**. New York: McGraw Hill; 1997.
4. Langley P, Simon HA, Bradshaw GL, Żytkow JM: **Scientific Discovery**. Cambridge, MA: MIT Press; 1987.
5. Džeroski S, Todorovski L (Eds): **Computational Discovery of Scientific Knowledge**. Berlin: Springer; 2007.
6. Džeroski S, Todorovski L: **Discovering dynamics: From inductive logic programming to machine discovery**. *J Intell Inf Syst* 1995, **4**:89-108.
7. Russel S, Norvig P: **Artificial Intelligence: A Modern Approach**. Second Edition. Upper Saddle River, NJ: Prentice Hall; 2003.
8. Todorovski L, Džeroski S: **Integrating domain knowledge in equation discovery**. In *Computational Discovery of Scientific Knowledge.* Edited by Džeroski S, Todorovski L. Berlin: Springer; 2007:69-97.
9. Falkenheiner B, Forbus KD: **Compositional modeling: Finding the right model for the job**. *Artif Intell* 1991, **51**:95-143.
10. Todorovski L, Ljubič P, Džeroski S: **Inducing polynomial equations for regression**. *Lect Notes Comput Sci* 2004, **3201**:441-452.
11. Džeroski S, Todorovski L, Ljubič P: **Using constraints in discovering dynamics**. *Lect Notes Comput Sci* 2003, **2843**:297-305.
12. Pečkov A, Džeroski S, Todorovski L: **A minimal description length scheme for polynomial regression**. *Lect Notes Comput Sci* 2008, **5012**:284-295.
13. Grünwald PD: *The Minimum Description Length Principle.* Cambridge, MA: MIT Press; 2007.
14. Pečkov A, Džeroski S, Todorovski L: **Multitarget polynomial regression with constraints**. In *Proceedings of the ECML/PKDD International Workshop on Constraint-Based Mining and Learning.* Edited by Nijssen S, de Raedt L. Warsaw, Poland: Warsaw University; 2007:61-72.
15. Voit EO: **Computational analysis of biochemical systems**. Cambridge, UK: Cambridge University Press; 2000.
16. Todorovski L, Džeroski S: **Declarative bias in equation discovery**. In *Proceedings of the Fourteenth International Conference on Machine Learning.* Edited by Fisher DH. San Mateo, CA: Morgan Kaufmann; 1997:376-384.
17. Gennemark P, Wedelin D: Efficient algorithms for ordinary differential equation model identification of biological systems. *IET Syst Biol* 2007, **1**:120-129.
18. Arkin RP, Ross J: **Statistical construction of chemical reaction mechanisms from measured time-series**. *J Phys Chem* 1995, **99**:970-979.
19. Bunch DS, Gay DM, Welsch RE: **Algorithm 717: subroutines for maximum likelihood and quasi-likelihood estimation of parameters in nonlinear regression models** *ACM Trans Math Soft* 1993, **19**:109-130.
20. Dennis JE, Gay DM, Welsch RE: **Algorithm 573: NL2SOL—an adaptive nonlinear least-squares algorithm**. *ACM Trans Math Soft* 1981, **7**:369-383.

21. Atanasova N, Todorovski L, Džeroski S, Rekar Remec S, Recknagel F, Kompare B: **Automated modelling of a food web in lake Bled using measured data and a library of domain knowledge**. *Ecol Model* 2006, **194**:37-48.
22. Džeroski S, Todorovski L: **Encoding and using domain knowledge on population dynamics for equation discovery**. In *Logical and Computational Aspects of Model-Based Reasoning.* Edited by Magnani L, Nersessian NJ, Pizzi C. Dordrecht: Kluwer; 2002:227-247.
23. Todorovski L, Džeroski S: **Using domain specific knowledge for automated modeling**. *Lect Notes Comput Sci* 2003, **2810**:48-59.
24. Todorovski L, Džeroski S: **Integrating knowledge-driven and data-driven approaches to modeling**. *Ecol Model* 2006, **194**:3-13.
25. Langley P, Sanchez J, Todorovski L, Džeroski S: **Inducing process models from continuous data**. In *Proceedings of the Nineteenth International Conference on Machine Learning.* Edited by Sammut C, Hofmann A. San Mateo, CA: Morgan Kaufmann; 1997:347-354.
26. Bridewell W, Langley P, Todorovski L, Džeroski S: **Inductive process modeling**. *Mach Learn* 2008, **71**:132.
27. Langley P, Shiran O, Shrager J, Todorovski L, Pohorille A: **Constructing explanatory process models from biological data and knowledge**. *Artif Intell Med* 2006, **37**:191-201.
28. Torralba A, Yu K, Shen P, Oefner P, Ross J: **Experimental test of a method for determining causal connectivities of species in reactions**. *Proc Natl Acad Sci* 2003, **100**:1494-1498.
29. Todorovski L, Bridewell W, Shiran O, Langley P: **Inducing hierarchical process models in dynamic domains**. In *Proceedings of the Twentieth National Conference on Artificial Intelligence* Edited by Veloso MM, Kambhampati S. Pittsburgh, PA: AAAI Press; 2005:892-897.
30. Garrett SM, Coghill GM, Srinivasan A, King RD: **Learning qualitative models of physical and biological systems**. In *Computational Discovery of Scientific Knowledge.* Edited by Džeroski S, Todorovski L. Berlin: Springer; 2007:248-272.
31. Zupan B, Bratko I, Demšar J, Juvan P, Kuspa A, Halter JA, Shaulsky G: **Discovery of genetic networks through abduction and qualitative simulation**. In *Computational Discovery of Scientific Knowledge.* Edited by Džeroski S, Todorovski L. Berlin: Springer; 2007:228-247.
32. Zapatka M, Koch Y, Brors, B: **Ontological analysis and pathway modelling in drug discovery**. *J Pharm Med* 2008, **22**:99-105.
33. Ideker T, Thorsson V, Ranish JA, Christmas R, Buhler J, Eng JK, Bumgarner R, Goodlett DR, Aebersold R, Hood L: **Integrated genomic and proteomic analyses of a systematically perturbed metabolic network**. *Science* 2001, **292**:929-934.
34. Sontag E, Kiyatkin A, Kholodenko BN: **Inferring dynamic architecture of cellular networks using time series of gene expression, protein and metabolite data**. *Bioinformatics* 2004, **20**:1877-1886.
35. Koza JR, Mydlowec W, Lanza G, Yu J, Keane MA: **Automatic computational discovery of chemical reaction networks using genetic programming**. In *Computational Discovery of Scientific Knowledge.* Edited by Džeroski S, Todorovski L. Berlin: Springer; 2007:205-227.
36. Kikuchi S, Tominaga D, Arita M, Takahashi K, Tomita M: **Dynamic modeling of genetic networks using genetic algorithm and S-system**. *Bioinformatics* 2003, **19**:643-650.

Inference of pathways from metabolic networks by subgraph extraction

Karoline Faust[1], Jérôme Callut[2], Pierre Dupont[3], and Jacques van Helden[4]

[1,4]Laboratoire de Bioinformatique des Génomes et des Réseaux (BiGRe),
Université Libre de Bruxelles, Campus Plaine - CP263, Boulevard du Triomphe, 1050 Bruxelles, Belgium.
[2,3]UCL Machine Learning Group, Université catholique de Louvain,
B-1348 Louvain-la-Neuve, Belgium.
{kfaust,jacques.van.helden}@ulb.ac.be
jcal@info.ucl.ac.be
pierre.dupont@uclouvain.be

Abstract. In this work, we present different algorithmic approaches to the inference of metabolic pathways from metabolic networks. Metabolic pathway inference can be applied to uncover the biological function of sets of co-expressed, enzyme-coding genes.

We compare the kWalks algorithm based on random walks and an alternative approach relying on k-shortest paths. We study the influence of various parameters on the pathway inference accuracy, which we measure on a set of 71 reference metabolic pathways. The results illustrate that kWalks is significantly faster and has a higher sensitivity but the positive predictive value is better for the pair-wise k-shortest path algorithm. This finding motivated the design of a hybrid approach, which reaches an average accuracy of 72% for the given set of reference pathways.

Key words: metabolic pathway inference, kWalks, k-shortest paths

1 Introduction

The products of co-expressed genes are often involved in a common biological function. In particular, the metabolic response to nutrients is generally regulated at multiple levels, including transcriptional. One approach to understand the function of co-expressed enzymes is to uncover the biological pathways in which they participate. This is usually achieved by mapping reactions associated to enzyme-coding genes on pre-defined metabolic pathways, i.e. [5, 11]. However, this approach does not deal well with transverse pathways (a set of reactions mapping to several pathways) and it fails if reactions belong to a pathway not yet included in the pre-defined set of reference pathways. Another strategy has been to infer metabolic pathways by finding the shortest paths between two reactions (e.g. those catalyzed by two co-expressed enzymes). One problem with this approach is the presence of compounds involved in a large number of reactions (co-factors and side-compounds like H_2O, ATP, NADPH), which tend to be used as shortcuts to connect any pair of nodes. Thus, a naive path finding approach results in biochemically invalid pathways, which contain co-factors or side compounds as intermediates. In our previous work, we tested different strategies to overcome this problem. First, we excluded a selected subset of highly connected compounds from the network [12, 13]. However, the choice of the compounds to be excluded is an issue, since some among the highly connected compounds participate in pathways as valid intermediate (e.g. purine nucleotide biosynthesis). We therefore introduced weighted networks [3, 4], in order to penalize highly connected compounds without excluding them from the graph. This approach yielded satisfactory results for the two-end linear path finding. In the present work, we extend the approach to *multiple-end pathway inference*: taking as input a set of seed reactions, we extract a subgraph that connects "at best" those seed nodes, according to some relevance criteria. The resulting pathway can correspond to an already known pathway, but it can also be a variant or a combination of known pathways, or even a novel pathway.

Pathway inference is as a more flexible alternative to pathway mapping. For instance, it can be used to infer pathways from operons, co-expressed genes or gene fusion events. It may also be applied in metabolic reconstruction in order to suggest possible pathways from genomic data for organisms with unknown metabolism.

We thoroughly evaluated the pathway inference performance of three algorithms: the *pair-wise k-shortest paths algorithm*, the *kWalks* algorithm [1, 6] and a hybrid algorithm that combines the former two.

2 Materials and Methods

2.1 Metabolic graph

In order to infer metabolic pathways, we need to represent metabolic data as a graph. We selected MetaCyc [10], the well-curated tier of BioCyc [2], as our data source, and constructed a bipartite, directed graph from all small molecule entries and their associated reactions contained in the OWL file of MetaCyc (Release 11.0). The resulting graph consists of 4,891 compound nodes and 5,358 reaction nodes. As discussed in [3], the direction of a reaction depends on physiological conditions in an organism (substrate and product concentrations, temperature). Since our graph is composed of data obtained from several hundred organisms, we considered that each reaction can be traversed either in forward or in reverse direction. Consequently, each reaction was represented as a pair of nodes, for the forward and the reverse directions, respectively. To prevent the k-shortest paths algorithm to cross the same reaction twice, forward and reverse direction are mutually exclusive. After this duplication of reaction nodes, we obtain a directed graph with 15,607 nodes and 43,938 edges. From now on, we will refer to this graph as the *MetaCyc* graph.

2.2 Reference pathways

We obtained a selected set of 71 known *S. cerevisiae* pathways from BioCyc (Release 11.0). All pathways in this reference set consist of at least 5 nodes and are included in the largest connected component of the MetaCyc graph. On average, the pathways are composed of 13 nodes and in addition, more than half of them are branched and/or cyclic.

2.3 Algorithms

All algorithms tested here take as input the nodes of interest (termed seed nodes or seeds) as well as a weighted input graph, and return a subgraph that connects the seeds.

Pair-wise k-shortest paths This approach relies on repetitively calling a k-shortest paths algorithm. K-shortest paths algorithms enumerate all simple paths (paths containing each node only once) between a start and an end node in the order of their length. In a weighted graph, paths are listed in the order of their weight.
In the first step of pair-wise k-shortest paths, a k-shortest paths algorithm [7] is called successively on each pair of seed nodes. A k-shortest instead of a shortest paths algorithm is employed to ensure that all lightest paths between a seed node pair are collected. The resulting path sets are stored in a path matrix. The minimal weight between each node pair is stored in a distance matrix. For the undirected MetaCyc graph, these matrices are symmetric. For the directed MetaCyc graph, the reverse paths between two seeds can be obtained by reversing the order of path nodes and their reaction directions.
In the second step of the algorithm, the subgraph is constructed from the path sets, starting with the lightest path set. Step-wise, more path sets are merged with the subgraph in increasing order of their weight. The process stops if either all seeds belong to one connected component of the subgraph or all path sets have been merged with the subgraph. The resulting subgraph represents the inferred pathway.
This algorithm is time-consuming, since the number of calls to the k-shortest paths algorithm increases quadratically with the seed node number.

kWalks The key idea of kWalks is that some edges in the input graph are more relevant than others to connect the seed nodes. The relevance of an edge is measured as the expected number of times it is visited along random walks connecting seed nodes. These expected passage times can be obtained using basic Markov chain theory [8]. A transition probability matrix P is derived from the adjacency matrix of the graph using simple edge weight normalization. For each seed node x, the submatrix xP is defined by considering only the lines and columns of P corresponding to x and all non-seed nodes. The fundamental matrix $xN = (I - xP) - 1$ contains useful information for computing the desired expectation. The entry $xNxi$ gives the number of times node i has been visited during walks starting in node x and ending when any seed node (except x) is reached. The expected number of passage time $xE(i, j)$ on an edge $i \rightarrow j$ is obtained by multiplying $xNxi$ by the probability P_{ij} of transiting from node i to node j. Finally, the relevance of an edge $i \rightarrow j$ is given by averaging $xE(i, j)$ over all seed nodes x. This technique is time-consuming since it relies on matrix inversions, which are generally performed with a cubic time complexity in the number of nodes in the graph.

An alternative approach considers random walks of a bounded length, i.e. only walks up to a prescribed length are allowed. The passage time expectations during such walks can be computed in linear time with respect to the number of graph edges and the maximum walk length using forward-backward recurrences [1, 6]. Moreover, bounding the walk length controls the level of locality while connecting seed nodes, which can be useful for pathway recovery.

Once the edge relevance has been obtained, a subgraph can be extracted by adding edges in the order of their relevance with respect to seed nodes, until either all seed nodes are connected or all edges have been added.

The output of kWalks is a list of edge relevance values. We can replace the original edge weights by these relevances and iterate kWalks by re-launching it on the input graph with updated weights.

In contrast to pair-wise k-shortest paths, the pathways inferred by kWalks may contain branches ending in non-seed nodes. We remove these branches in a post-processing step.

Hybrid approach The hybrid approach combines kWalks with the pair-wise k-shortest paths algorithm. First, kWalks is launched to extract a fixed percentage of the input graph. The final pathway is then extracted from the kWalks subgraph using the pair-wise k-shortest paths algorithm.

2.4 Parameter combinations

The performance of kWalks, pair-wise k-shortest paths and the hybrid approach was evaluated with a number of different parameter values.

Iteration number For the kWalks and hybrid algorithm, we ran 1, 3 or 6 iterations of the kWalk algorithm.

Graph weight We weight the metabolic graph to avoid highly connected compounds. As in our previous work [3, 4], we assign to each compound node its degree as weight (compound degree weight) or use an un-weighted graph for comparison (unit weight). In addition, we test a weighting scheme where compound node weights are taken to the power of two (inflated compound degree weight). Since the pair-wise k-shortest paths and kWalks assume weights on edges rather than nodes, the initial degree-based node weights are transformed into edge weights by taking the mean of the weights of the nodes adjacent to an edge.

Re-use of kWalks edge relevances In the hybrid approach, we may either use the weights from the input graph or the edge relevances computed by kWalks to weight the extracted subgraph. In addition, when iterating kWalks, we may modify the edge relevances by inflating them (taking them to the power of a positive integer) to increase the difference between relevances. We tested all combinations resulting from these options.

Directionality In order to support reaction reversibility, we represent each reaction by two nodes, one for the direct and one for the reverse direction. In addition, we also constructed an undirected version of the MetaCyc graph, where each reaction is represented by only one node, which is connected to compound nodes by undirected edges.

Fixed subgraph extraction In the hybrid approach, after the last kWalks iteration we extract a subgraph of fixed size from the input graph. The size of this subgraph has been varied from 0.1% to 10% of the edges ranked by relevance. The subgraph obtained by fixed size extraction may consist of more than one component.
The subgraph size optimization has been performed in the directed, compound-weighted MetaCyc graph without iterating kWalks or inflating edge relevances. The input graph weights rather than the edge relevances were fed into the second step of the hybrid algorithm.

2.5 Evaluation procedure

For each pathway, several inferences are tested, with increasing seed node number, in order to test the impact of the seed number on the accuracy of the result. For each of the 71 reference pathways, we first select the terminal reactions as seeds, we infer a pathway that interconnects them, and we compare the nodes of the inferred pathways with those of the annotated pathway. Then, we progressively increase the number of seed reactions by adding randomly selected nodes of the reference pathway, and re-do the inference and evaluation, until all reactions of the pathway are selected as seeds.
We define as one experiment the set of all the pathway inferences performed for a given parameter value combination (e.g. pair-wise k-shortest paths on directed graph with compound node weights). We did 82 such experiments to find the optimal parameter value combination for each algorithm.

Scores The accuracy of an inferred pathway is calculated based on the correspondence between its non-seed nodes and those of the reference pathway. We define as true positive (TP) a non-seed node that is present in the reference as well as the inferred pathway. A false negative (FN) is a non-seed node present in the reference but missing in the inferred pathway and a false positive (FP) is a non-seed node absent in the reference but found in the inferred pathway. The sensitivity (Sn) is defined as the ratio of inferred true instances versus all true instances, whereas the positive predictive value (PPV) gives the ratio of inferred true instances versus all inferred instances.

$$Sn = \frac{TP}{(TP + FN)} \tag{1}$$

$$PPV = \frac{TP}{(TP + FP)} \tag{2}$$

We can combine sensitivity and positive predictive value to calculate the accuracy as their geometric mean.

$$Acc_g = \sqrt{Sn * PPV} \tag{3}$$

3 Results

3.1 Study case aromatic amino acid biosynthesis

To illustrate the idea of pathway inference, we will discuss the aromatic amino acid biosynthesis pathway. Figure 1A shows the pathway as annotated in BioCyc. This pathway is active in *E. coli* and produces aromatic amino acids (tyrosine, tryptophan and phenylalanine) from erythrose-4-phosphate. The first part of this pathway is linear and ends in chorismate. From chorismate onwards, the pathway splits into three branches, one leading to tryptophan and the other bifurcating to phenylalanine and tyrosine respectively. The entire pathway, excluding the terminal compounds, is made up of 34 compound and reaction nodes.

The aromatic amino acid pathway is tightly regulated on the transcriptional level. In presence of one of the end products (that is an aromatic amino acid), the corresponding synthesis branch is down-regulated. The linear part of the pathway is also subject to regulation (on the enzymes catalyzing the first, fifth and sixth reaction) integrating feed-back loops from the three end-products. From the set of transcriptionally regulated reactions, we selected DAHPSYN-RXN, SHIKIMATE-KINASE-RXN, PRAISOM-RXN, PHEAMINOTRANS-RXN, TYRAMINOTRANS-RXN and RXN0-2382 (BioCyc identifiers) as seed nodes. In our previous 2-end path finding approach [3, 4] we were restricted to only two seed nodes. To simulate this situation, we applied the pair-wise k-shortest paths algorithm on the start (DAHPSYN-RXN) and one of the end reactions (RXN0-2382). The resulting pathway, shown in Figure 1B, connects the two seed reactions via a shortcut, bypassing a major part of the reference pathway. The resulting linear path fits the branched reference pathway with a low accuracy (14%). However, if we repeat the inference with the full seed node set, we recover the reference pathway with an accuracy of 97% (Figure 1C). Thus, without surprise, we observe that multi-seed subgraph extraction is more appropriate to infer branched pathways than 2-end path finding. In the next section, we evaluate various algorithms and parametric choices for the multi-seed subgraph extraction.

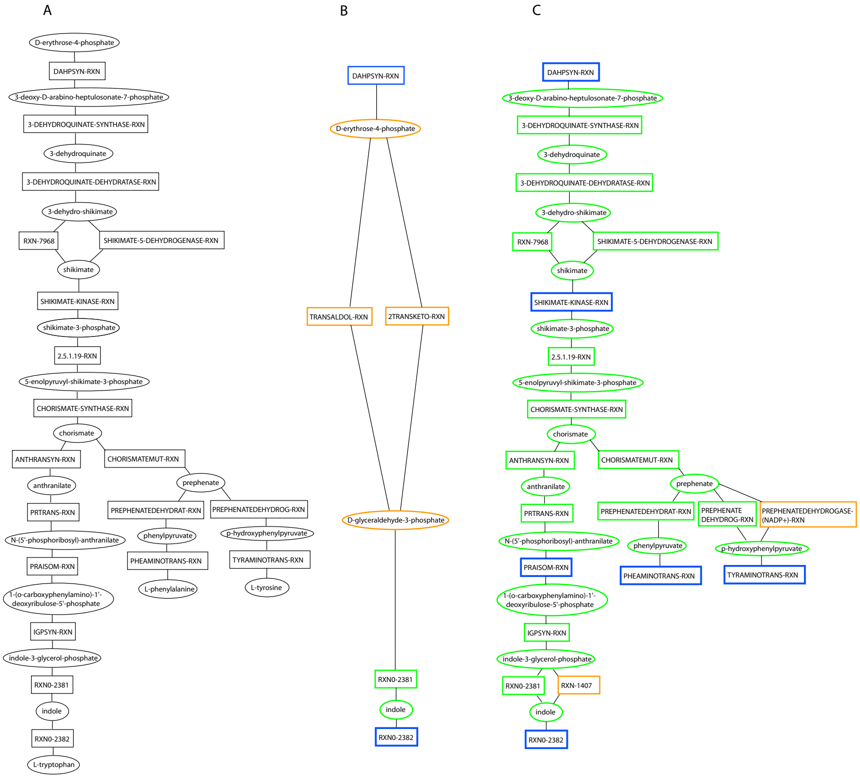


Fig. 1. Aromatic amino acid biosynthesis pathway as annotated in BioCyc (A), inferred with two seed reactions (B) and inferred with 6 seed reactions (C). For both inferences, the pair-wise k-shortest paths algorithm has been run on the compound-weighted, directed MetaCyc graph. Seed reactions have a blue border, false positive nodes an orange border and true positive nodes a green border. Compound nodes are represented as ellipses and labeled with their names, whereas reaction nodes are drawn as rectangles and labeled with their BioCyc identifiers.

3.2 Parameter combinations

The performance of the pair-wise k-shortest paths algorithm, kWalks and the hybrid approach has been evaluated for 82 parameter value combinations ac-

Table 1. The ten pathway inference experiments (parametric combinations) resulting in the highest geometric accuracy in our evaluation. For each experiment, its parameter values, its runtime and its geometric accuracy (averaged over all inferences) are displayed.

Algorithm	Iteration number	Weighting scheme	Inflation after iteration	Directed graph	Edge relevances used as weight	Geometric accuracy	Runtime in seconds
Hybrid	6	Compound degree weight	False	True	False	0.6822	636
Pair-wise k-shortest paths	0	Compound degree weight	False	True	False	0.6803	445
kWalks	3	Compound degree weight	True	True	False	0.6796	130
kWalks	6	Inflated compound degree weight	False	True	False	0.679	309
Hybrid	3	Compound degree weight	False	True	False	0.6786	393
kWalks	6	Compound degree weight	True	True	False	0.6778	323
kWalks	6	Compound degree weight	False	True	False	0.6773	312
Hybrid	0	Compound degree weight	False	True	False	0.6757	183
Hybrid	6	Compound degree weight	True	True	False	0.6738	744
Hybrid	3	Compound degree weight	True	True	False	0.6724	431

cording to the Sn, PPV and Acc_g criteria described above.

Table 1 lists the top ten experiments, with geometric accuracies averaged over all inferences done for each experiment. Interestingly, all three algorithms are present in Table 1. In agreement with our previous analysis [3, 4], directed, compound-weighted graphs yield highest pathway inference accuracies. The performance of the hybrid approach increases if the original graph weights rather than the edge relevances obtained by kWalks are fed into the pair-wise k-shortest paths algorithm. Inflating edge relevances after kWalks iterations does not have a significant impact on pathway inference accuracy.

It is worth noting that kWalks without iteration is not among the top ten experiments. Figure 2A shows a summary of all inferences done for kWalks without iteration in the directed, un-weighted MetaCyc graph. For comparison, Figure 2B displays the geometric accuracies obtained for kWalks with three iterations under the same conditions. Closer inspection of the geometric accuracy heat maps reveals that iterating kWalks improves geometric accuracies for the tryptophan biosynthesis, isoleucine biosynthesis I and UDP-N-acetylgalactosamine biosynthesis pathways. The average geometric accuracy of kWalks increases from 62% for one iteration to 64% for three iterations. This illustrates that calling kWalks iteratively improves pathway inference accuracy. Not surprisingly, Table 1 lists only kWalks experiments in which this algorithm has been iterated three or six times.

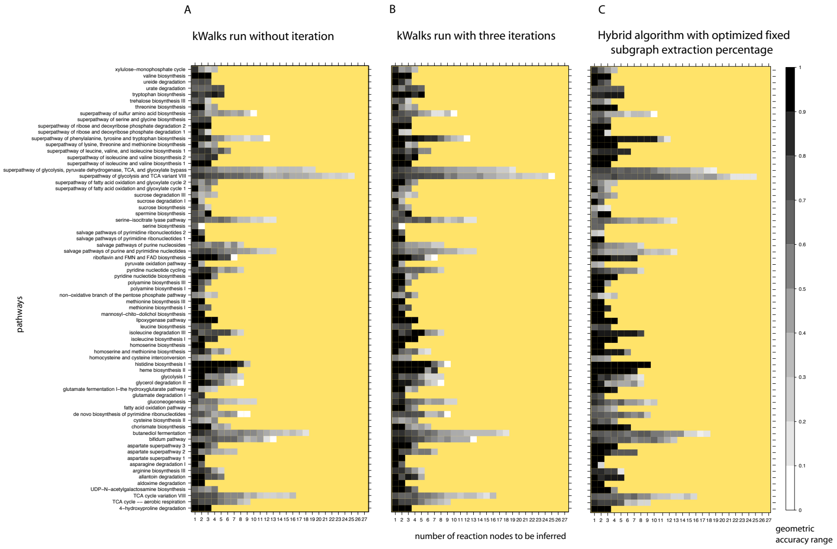


Fig. 2. The geometric accuracies obtained for each pathway as a heat map. The x-axis indicates the number of non-seed reaction nodes (those that the algorithm needs to infer). Each row corresponds to one reference pathway. The geometric accuracy is reflected by a gray scale from white ($Acc_g = 0$) to black ($Acc_g = 1$). A. Heat map obtained for kWalks in the un-weighted, directed MetaCyc graph without inflation or iteration. B. Heat map obtained under the same conditions, but with kWalks iterated three times. C. Summary of the inferences done for the hybrid approach in the directed, compound-weighted MetaCyc graph. The size of the subgraph extracted by kWalks was set to 0.5%, which was the optimal value found by our evaluation.

3.3 Hybrid algorithm optimization

In the evaluation shown in Table 1, we set the fixed subgraph size for the hybrid approach to 5%. However, subsequent variation of the subgraph size in a percentage range from 0.1% to 10% showed that 0.5% is the optimal percentage for subgraph extraction in the hybrid approach. When the subgraph extraction percentage is set to 0.5%, the average geometric accuracy of the hybrid approach reaches 72%, which is the highest percentage obtained for any experiment. Figure 2C shows the geometric accuracy of each inference in a heat map.

4 Discussion

4.1 Parameters

Graph directionality The directed MetaCyc graph yields higher geometric accuracies than the undirected one. In the undirected MetaCyc graph, it is possible to traverse the graph from substrate to substrate or from product to product, which is prevented in the directed graph.

Compound weighting In our previous studies [3, 4], we showed that the weight is the most determinant parameter for inferring relevant pathways by 2-end path finding. As expected, node weighting also exerts a strong impact on the performances of multi-seed pathway inference, but this impact depends on the algorithm used. The pair-wise k-shortest paths performed best in the compound-weighted MetaCyc graph. But surprisingly, kWalks without iteration resulted in higher accuracies when applied to the un-weighted rather than to the weighted MetaCyc graph. Interestingly, kWalks automatically induces weights that favor relevant compounds. Actually, the kWalks relevance score can be interpreted as a context-specific betweenness index, and we can thus understand that it penalizes highly connected compounds, thereby explaining the good results obtained by relevance weighting. In the hybrid approach, the resulting average geometric accuracy for the un-weighted graph is higher when we pass the kWalks induced weights (edge relevances) rather than the original weights to the second step of the algorithm. However, if we run the hybrid approach on the weighted graph, the kWalks induced weights decrease the accuracy compared to the original weights. Inflation of kWalks induced weights does not improve results significantly.
For most parameters, we determined optimal values and their combination (with respect to the reference pathways) by an exhaustive search. However, compound weights were chosen heuristically and may be optimized in future by a machine learning approach.

4.2 Algorithms

Although the pair-wise k-shortest paths algorithm is slow (7 minutes per pathway inference in average), its average geometric accuracy figures among the top ten. In contrast, kWalks without iteration runs in seconds, but yields unsatisfactory accuracies. Upon closer inspection, it became apparent that kWalks results in high sensitivities and low positive predictive values. The positive predictive value of the kWalks algorithm can be increased by invoking it iteratively or by combining it with pair-wise k-shortest paths in the hybrid algorithm. Both

strategies are paid with a longer runtime.
The highest average geometric accuracy was obtained with the optimized hybrid approach. This shows that pair-wise k-shortest paths and kWalks are complementary. The focus of kWalks is to capture the part of the input graph most relevant for connecting the seeds, reducing the number of false negatives at the cost of increasing the number of false positives. Pair-wise k-shortest paths can then discard false positives introduced by kWalks.

4.3 Similar approach to subgraph extraction

Recently, Koren and co-workers [9] designed a proximity measure that avoids dead-end nodes and takes node degree and multiple paths between seeds into account. Based on this measure, they describe a subgraph extraction approach that relies on finding the k shortest paths between seed nodes. The resulting paths are combined in such a way that proximity between seeds is maximized while minimizing the subgraph size.
This extraction procedure aims at capturing the paths that contribute to the proximity of nodes. It is worth noting that two nodes with many paths between them are considered closer than two nodes connected by a few paths. Therefore, this algorithm will likely return more alternative paths between seeds than our algorithms do. This is very interesting when one wants to explore the metabolic neighborhood of a set of seed nodes, but less desirable when a metabolic pathway should be predicted.

5 Conclusion

We have presented three different algorithmic approaches to infer metabolic pathways from metabolic graphs: kWalks, pair-wise k-shortest paths, and a hybrid that combines both.
The former two algorithms have complementary strengths and weaknesses. Our evaluation on 71 yeast pathways has shown that their combination in the hybrid approach yields the highest geometric accuracies.

In future, we will apply these algorithms to microarray data to infer metabolic pathways from co-expressed enzyme-coding genes.

Acknowledgments

KF is supported by Actions de Recherches Concertes de la Communaut Franaise de Belgique (ARC grant number 04/09-307). The BiGRe laboratory is a member of the BioSapiens Network of Excellence funded under the sixth Framework program of the European Communities (LSHG-CT-2003-503265). The work was supported by the Belgian Program on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office, project P6/25 (BioMaGNet).

We would like to thank the referees for their helpful comments, in particular for pointing to the article of Koren and co-workers.

References

1. Callut, J.: First Passage Times Dynamics in Markov Models with Applications to HMM Induction, Sequence Classification, and Graph Mining. PhD Thesis Dissertation, Université catholique de Louvain (2007)
2. Caspi, R. et al.: The MetaCyc Database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases. Nucleic Acids Research 36, D623–D631 (2008)
3. Croes, D., Couche, F., Wodak, S., van Helden, J.: Metabolic PathFinding: inferring relevant pathways in biochemical networks. Nucleic Acids Research 33, W326–W330 (2005)
4. Croes, D., Couche, F., Wodak, S., van Helden, J.: Inferring Meaningful Pathways in Weighted Metabolic Networks. J. Mol. Biol. 356, 222–236 (2006)
5. Dahlquist, K.D., Salomonis, N., Vranizan, K., Lawlor, S.C., Conklin, B.R.: GenMAPP, a new tool for viewing and analyzing microarray data on biological pathways. Nature Genetics 31, 19–20 (2002)
6. Dupont, P., Callut, J., Dooms, G., Monette, J.-N., Deville, Y.: Relevant subgraph extraction from random walks in a graph. Research Report UCL/FSA/INGI RR 2006-07 (2006-07)
7. Jimenez, V.M., Marzal, A.: Computing the K-shortest Paths: a New Algorithm and an Experimental Comparison. In: Proc. 3rd Int. Worksh. Algorithm Engineering (WAE 1999) 1668, pp. 15–29 (1999)
8. Kemeny, J.G., Snell, J.L.: Finite Markov Chains. Springer-Verlag (1983)
9. Koren, Y., North, S.C., Volinsky, C.: Measuring and extracting proximity in networks, In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 245–255 (2006)
10. Krieger, C. J. et al.: MetaCyc: a multiorganism database of metabolic pathways and enzymes. Nucleic Acids Research 32, D438–D442 (2004)
11. Paley, S.M., Karp, P.D.: The Pathway Tools cellular overview diagram and Omics Viewer. Nucleic Acids Research 34, 3771–3778 (2006)
12. van Helden, J., Gilbert, D., Wernisch, L., Schroeder, M., Wodak, S.: Applications of regulatory sequence analysis and metabolic network analysis to the interpretation of gene expression data. In: Lecture Notes in Computer Science 2066, pp. 155–172 (2001)
13. van Helden, J., Wernisch, L., Gilbert, D., Wodak, S.: Graph-based analysis of metabolic networks. In: Ernst Schering Res Found Workshop, pp. 245-74. Springer-Verlag (2002)

Predicting gene essentiality from expression patterns in *Escherichia coli*

Alexandre Irrthum[2] and Louis Wehenkel[1,2]

[1] Department of Electrical Engineering and Computer Science
[2] GIGA-Research, University of Liège, B4000 Belgium

Abstract. Essential genes are genes whose loss of function causes lethality. In the case of pathogen organisms, the identification of these genes is of considerable interest, as they provide targets for the development of novel antibiotics. Computational analyses have revealed that the positions of the encoded proteins in the protein-protein interaction network can help predict essentiality, but this type of data is not always available. In this work, we investigate prediction of gene essentiality from expression data only, using a genome-wide compendium of expression patterns in the bacterium *Escherichia coli*, by using single decision trees and random forests. We first show that, based on the original expression measurements, it is possible to identify essential genes with good accuracy. Next, we derive, for each gene, higher level features such as average, standard deviation and entropy of its expression pattern, as well as features related to the correlation of expression patterns between genes. We find that essentiality may actually be predicted based only on the two most relevant ones among these latter. We discuss the biological meaning of these observations.

1 Introduction

Robustness or fault-tolerance is one of the defining qualities of biological organisms. For example, genome-scale gene deletion studies in yeasts and bacteria have demonstrated that most of the genes are not essential for their growth and reproduction. The identification of the essential genes is of great theoretical and practical interest. From a theoretical standpoint, these studies are necessary for the identification of the “minimal genome”, the smallest set of genes that allows an organism to survive and reproduce. From a more practical point of view, the identification of essential genes in pathogen micro-organisms is a useful first step in the development of novel antibiotics [10].

Because the experimental identification of these genes is a costly and time-consuming process, methods for their computational identification have been proposed. Most notably, it has been shown that the positions of the encoded proteins in the protein-protein interaction network are good predictors of essentiality, with “hub” and “bottleneck” proteins being more often essential [9][14]. In another study, reduced stochastic fluctuation of expression has been associated with essential genes [6]. Finally, the importance of gene sequence features has also been demonstrated [13].

In this work, we investigate if essentiality can be predicted from gene expression patterns only, in the bacterium *Escherichia coli*. This is a first step in the assessment of classifiers based on expression patterns to infer gene essentiality across species.

2 Data and Methods

2.1 Data

A gene expression dataset for the bacterium *Escherichia coli* was obtained from the Many Microbes Database (http://m3d.bu.edu, [4]). This dataset contains expression data for 4217 genes across 305 experiments corresponding to various growth conditions and mutations. Some of the experiments were replicated, giving a total of 612 expression values for each gene. These expression data have been uniformly normalized with the RMA algorithm [8], making them comparable across conditions.

We also obtained gene essentiality data from the Keio Collection of *E. coli* gene knockouts [1]. In this experiment, 4288 genes were systematically inactivated by targeted deletion. The 303 genes for which it was impossible to obtain deletion mutants were identified as essential genes.

We work with the 4217 genes that are represented in both datasets, out of which 289 are identified as essential.

2.2 Feature generation

In addition to directly using the expression vectors for classification, we extracted 38 higher level features for each gene in the dataset. Features F1 to F4 are based on the expression patterns of genes considered individually, while features F5-F38 are based on the similarities between the expression patterns of a gene and other genes in the dataset.

Individual features

F1: Mean gene expression level across $n = 305$ experimental conditions

$$mean = \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i,$$

where x_i is the average expression level of the gene over the repeated experiments of condition i.

F2: Standard deviation of gene expression level across $n = 305$ experimental conditions

$$std = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}.$$

F3: Shannon entropy of gene expression level across $n = 305$ experimental conditions [12]

$$entropy = -\sum_{i=1}^{n} p_i \; log \; p_i, \text{ where } p_i = \frac{x_i}{\sum_{i=1}^{n} x_i}.$$

F4: Mean within-repeats standard deviation

$$repeat\_std = \frac{1}{m}\sum_{i=1}^{m}\sqrt{\frac{1}{r_i - 1}\sum_{j=1}^{r_i}(x_{ij} - x_i)^2},$$

where $m = 177$ is the number of experimental conditions with repeats, r_i is the number of repeats (2 or 3) for condition i, x_{ij} is the expression level of the gene for repeat j of condition i and x_i is the mean expression level of the gene in condition i.

Global features

We use Pearson's correlation coefficient to compute similarities between two gene expression vectors $\mathbf{x}$ and $\mathbf{y}$:

$$corr(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}.$$

The matrix of pairwise correlations between the 4217 genes was computed. Correlation coefficients were transformed into dissimilarities with $diss(\mathbf{x}, \mathbf{y}) = (1 - corr(\mathbf{x}, \mathbf{y}))/2$ to give values in the range $[0, 1]$.

F5-F17: Dissimilarity of the gene expression pattern with the expression pattern of its k-th nearest neighbor for $k \in$(1, 2, 5, 10, 20, 50, 100, 200, 500, 1000, 2000, 4000, 4216).

Next, we computed a new measure of pairwise similarity between genes, based on the notion of shared nearest neighbor (SNN, [3]). The algorithm for the computation of shared nearest neighbor similarities is:

```
Find the k-nearest neighbors of each gene
For every pair of genes (x, y):
    if x and y are not amongst each other's k-nearest neighbors:
        similarity(x, y) <- 0
    else:
        similarity(x, y) <- number of shared neighbors
    endif
```

We generated SNN similarity matrices for k ∈(10, 20, 50, 100, 200, 500, 1000). The parameter k controls the sparsity of the resulting SNN similarity matrices, with smaller k producing sparser matrices where only highly correlated genes have non-null similarities. These similarity matrices can be represented as graphs, where two similar genes are connected by an edge whose weight is the similarity. From these similarity graphs, three additional features were extracted for each gene; degree, sum of edge weights and betweenness centrality.

F18-F24: Degree of the gene for k ∈(10, 20, 50, 100, 200, 500, 1000)

$$degree\_k(gene) = number\ of\ connections\ incident\ to\ the\ gene.$$

F25-F31: Sum of edge weights for k ∈(10, 20, 50, 100, 200, 500, 1000)

$$weight\_k(gene) = sum of the weights of connections incident to the gene.$$

F32-F38: Betweenness centrality for k ∈(10, 20, 50, 100, 200, 500, 1000)

$$betweenness\_k(gene) = \sum_{s \neq gene \neq t \in V} \frac{\sigma_{st}(gene)}{\sigma_{st}},$$

where σ_{st} is the total number of shortest paths between genes s and t, and $\sigma_{st}(gene)$ is the number of those shortest paths that also pass through the gene under consideration [7]. Betweenness centrality values were computed with the Boost graph library (http://www.boost.org).

2.3 Machine Learning

The gene essentiality data set is highly unbalanced, with 3928 non-essential genes for only 289 essential genes. We therefore learned the classifiers on a balanced dataset obtained by completing the 289 essential genes with a random sample of 289 non-essential ones.

For the classification of genes with respect to essentiality, we used random forests [2] with 10-fold cross-validation, implemented in the Weka

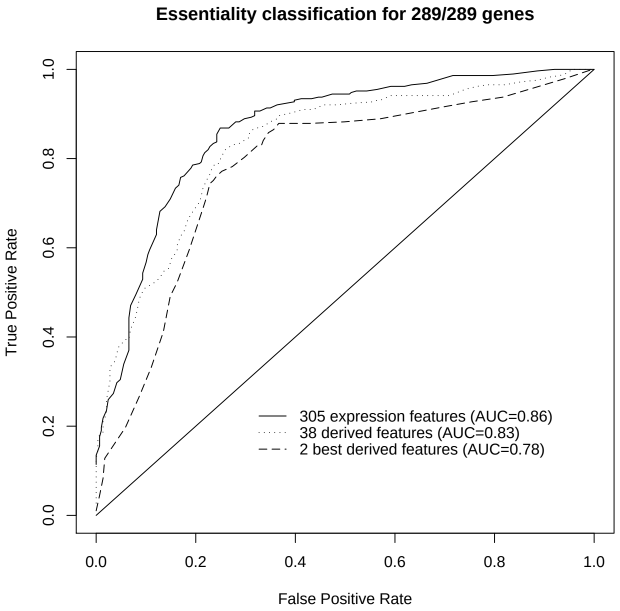


Fig. 1. ROC curves for the class of essential genes, with 10-fold cross validation on the balanced dataset. AUC: Area under the ROC curve.

.

machine learning workbench [5]. Hundred trees were constructed for each fold, each based on a random selection of 9 attributes at each node (from the 305 expressions), or 6 attributes (from the 38 derived features). The number of randomly selected attributes at each node is the largest integer less than $log_2(m) + 1$, where m is the number of input attributes [2].

We also used J4.8, a variant of the C4.5 decision tree classifier [11] implemented in Weka, to determine classification performance based on the two most important features, *mean* and *weight_1000* (the importances were derived from the random forest classifier).

3 Results and discussion

Figure 1 shows the ROC curves for the class of essential genes, based on 10-fold cross-validation on the balanced dataset. The upper ROC curve is for random forests predictions based on the gene expressions over 305 conditions. The middle curve is for random forests predictions based on the 38 derived features. Finally, the bottom ROC curve corresponds to classification performance with a single J4.8 decision tree using the two most important features derived by the random forests classifier, *mean* and *weight_1000*. These ROC curves indicate that it is possible to achieve a high precision for the detection of a subset of essential genes, with all three approaches. The AUC values show that most of the relevant information contained in the gene expressions is kept in the set of high level feature, and that the two most significant ones among them already allow to predict gene essentiality quite well.

Table 1. Classifier performances on balanced dataset (10-fold cross-validation) and, rightmost column, on the non-essential genes not present in balanced dataset (classifier trained on balanced dataset). ER: Error rate.

Classifier	ER	Precision	Recall	AUC	ER-test
305_exp	20.24%	77.56%	83.74%	0.865	21.19%
38_feat	23.52%	77.22%	75.09%	0.832	21.05%
2_best	24.22%	76.51%	74.39%	0.779	14.20%

Table 1 gives a summary of the performances of the three classifiers, first on the balanced training dataset itself (using a stratified 10-fold cross-validation), and second on a testing dataset comprising the 3639 non-essential genes not present in the training dataset. We observe that the classifiers derived from the balanced dataset, which use only a small fraction of the non-essential genes, maintain their accuracy on the testing dataset.

The J4.8 decision tree obtained by using the two features is as follows:

```
mean <= 8.625629: non_essential (219/35) (leaf 1)
mean > 8.625629
|   weight_1000 <= 716748
|   |   weight_1000 <= 447278: non_essential (77/32) (leaf 2)
|   |   weight_1000 > 447278
|   |   |   mean <= 9.199547: non_essential (44/21) (leaf 3)
|   |   |   mean > 9.199547: essential (107/25) (leaf 4)
|   weight_1000 > 716748: essential (131/12) (leaf 5)
```

where the numerical values, e.g. (219/35) for the first terminal node, give the number of genes reaching the node, and the number of them that are misclassified, e.g. 35 essential genes classified as non-essential at the first node. Figure 2 illustrates its classification over the genes in the training set, with the delimited areas corresponding to the five terminal nodes of the tree. A preliminary analysis shows that among 36 essential genes encoding proteins of the 30S and 50S ribosomal subunits (families *rps*

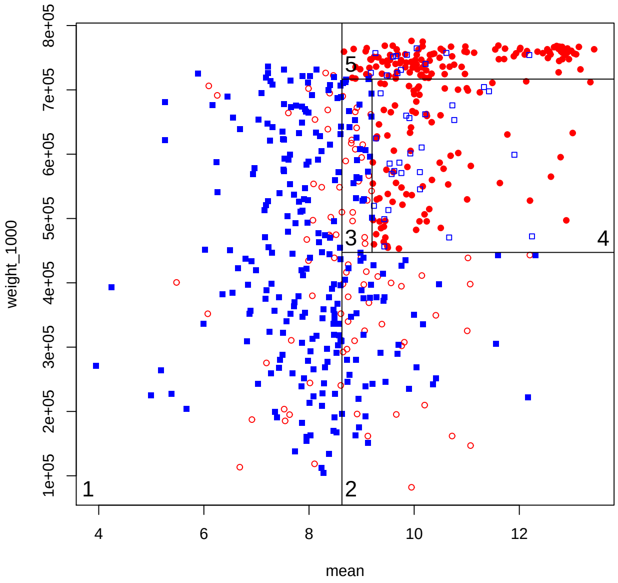


Fig. 2. Plot of the 578 genes in balanced dataset. Numbers 1 to 5 correspond to the numbering of the leaves in the tree. Code: circle and red: essential gene; square and blue: non-essential gene; filled symbol: correct classification; open symbol: misclassification

and *rpl*), 32 are in the upper right area of the graph (area 5). It will be interesting to determine if, more generally, essential genes present in different regions of the plot correspond to particular classes of biological functions.

This decision tree has a meaningful biological interpretation. First, we observe that genes with high mean expression levels are more often essential. In this experiment, expression is measured at the level of the mRNA and thus is related to the rate of production of the corresponding protein. It is not surprising that proteins that are more actively synthesized should be more vital to the bacterium. Second, we observe that genes densely connected to other genes based on the correlation of their expression patterns (the *weight_1000* feature) are also more often essential. This observation parallels the known correlation between connectivity in protein-protein interaction networks and essentiality [9]. Sets of genes with highly correlated expression patterns often belong to a same basic molecular complex, such as the RNA or the protein synthesis machineries, and such complexes are central to the functioning of the cell.

4 Conclusion

In this paper we have explored the prediction of gene essentiality from mRNA expression patterns, by applying tree-based machine learning methods on an experimental dataset of genes from *Escherichia coli*.

Our work shows that it is indeed possible to predict essential genes based solely on expression patterns and, importantly, that this may be achieved by using only a couple of high-level *global* features and a very simple decision tree.

Future analyses will have to compare the prediction based on gene expressions with the prediction based on protein-protein interaction data. In a second stage, we aim at building classification models for other species. The end-goal of this research is to develop classifiers that allow to infer gene essentiality across species, so as to exploit experimental data from some species to predict essentiality of genes of other species.

Acknowledgments

The authors would like to thank the two anonymous reviewers and Pierre Geurts for their insightful comments. This paper presents research results of the Belgian Network BIOMAGNET (Bioinformatics and Modeling: from Genomes to Networks), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with the authors.

References

1. Baba T., Ara T., Hasegawa M., Takai Y., Okumura Y., Baba M., Datsenko K.A., Tomita M., Wanner B.L., Mori H.: Construction of Escherichia coli K-12 in-frame, single-gene knockout mutants: the Keio collection. *Molecular Systems Biology* 2:2006.0008 (2006)
2. Breiman L.: Random forests. *Machine Learning* 45(1):5-32 (2001)
3. Ertoz L., Steinbach M., Kumar V.: A new shared nearest neighbor clustering algorithm and its applications. In *Workshop on Clustering High Dimensional Data and its Applications, Proc. of Text Mine 01, First SIAM Intl. Conf. on Data Mining* (2001)
4. Faith J.J., Driscoll M.E., Fusaro V.A., Cosgrove E.J., Hayete B., Juhn F.S., Schneider S.J., Gardner T.S.: Many Microbe Microarrays Database: uniformly normalized Affymetrix compendia with structured experimental metadata. *Nucleic Acids Research* 36(Database issue):D866-70 (2008)
5. Frank E., Hall M., Trigg L., Holmes G., Witten I.H.: Data mining in bioinformatics using Weka. *Bioinformatics* 20(15):2479-81 (2004)
6. Fraser H.B., Hirsh A.E., Giaever G., Kumm J., Eisen M.B.: Noise minimization in eukaryotic gene expression. *PLoS Biology* 2(6):0834-0838 (2004)

7. Freeman L.C.: A Set of measures of centrality based upon betweenness. *Sociometry* 40:35-41 (1977)
8. Irizarry R.A., Bolstad B.M., Collin F., Cope L.M., Hobbs B., Speed T.P.: Summaries of Affymetrix GeneChip probe level data. *Nucleic Acids Research* 31(4):e15 (2003)
9. Jeong H., Mason S.P., Barabsi A.L., Oltvai Z.N.: Lethality and centrality in protein networks. *Nature* 411(6833):41-2 (2001)
10. Pucci M.J.: Use of genomics to select antibacterial targets. *Biochemical Pharmacology* 71(7):1066-72 (2006)
11. Quinlan J.R.: *C4.5: Programs for Machine Learning.* Morgan-Kaufmann Publishers, San Mateo, CA. (1993)
12. Schug J., Schuller W.P., Kappen C., Salbaum J.M., Bucan M., Stoeckert C.J. Jr.: Promoter features related to tissue specificity as measured by Shannon entropy. *Genome Biology* 6:R33 (2005)
13. Seringhaus M., Paccanaro A., Borneman A., Snyder M., Gerstein M.: Predicting essential genes in fungal genomes. *Genome Research* 16:1126-1135 (2006)
14. Yu H., Kim P.M., Sprecher E., Trifonov V., Gerstein M.: The importance of bottlenecks in protein networks: correlation with gene essentiality and expression dynamics *PLoS Computational Biology* 3(4):e59 (2007)

A Structured-Outputs Method for Prediction of Protein Function

Artem Sokolov and Asa Ben-Hur

Colorado State University, Fort Collins CO 80523, USA,
{sokolov, asa}@cs.colostate.edu

Abstract. Despite several decades of research, the predominant approach to prediction of protein function follows the "transfer-of-annotation" paradigm, where a query protein is compared against a database and annotated with the function of similar proteins. The Gene Ontology terms used to annotate protein function form a hierarchy of thousands of terms making standard multi-class or multi-label approaches impractical. We propose a method for modeling the structure of the Gene Ontology hierarchy in the framework of structured-output methods and present a structured-perceptron algorithm for predicting protein function. Our empirical results demonstrate that learning the structure of the output space yields better performance when compared to the traditional transfer-of-annotation methodology.

1 Introduction

We address the problem of automatic annotation of protein function using structured output methods. The function of a protein is defined by a set of keywords that specify its molecular function, its role in the biological process and its localization to a cellular component. The Gene Ontology (GO), which is the current standard for annotating gene products and proteins, provides a large set of terms arranged in a hierarchical fashion [7].

Computational methods for annotating protein function have been predominantly following the "transfer-of-annotation" paradigm where GO keywords are transferred from one protein to another based on the sequence similarity between the two. This is generally done by employing a sequence alignment tool such as BLAST [1] to find annotated proteins that have a high level of sequence similarity to an un-annotated query protein. Such variations on the nearest-neighbor methodology suffer from serious limitations in that they fail to exploit the inherent structure of the annotation space. Furthermore, annotation transfer of multiple GO keywords between proteins is not always appropriate, e.g. in the case of multi-domain proteins [6].

Prediction of protein function has also been approached as a binary classification problem using a wide array of methods that predict whether a query protein has a certain function [5, 10, 19, 11]. Alternatively one can learn to recognize "good" BLAST hits, from which annotations can then be transferred, an approach taken by Vinayagam, *et al.* [21]. These methods leave it to the user to combine the output of classifiers trained to recognize the different possible functions and decide which of the annotations to accept.

Since proteins can have multiple functions, and those functions are described by a hierarchy of keywords, prediction of protein function can be formulated as a hierarchical multi-label classification problem. Barutcuoglu, *et al.* [3] adopted an approach, where a classifier is trained for each node in the ontology and the classifier outputs are combined through the use of a Bayesian network to infer the most likely set of consistent labels. This is a step towards learning the structure of the output space. Structured-output methods are a generalization of this approach where a single classifier is trained to predict the entire structured label [2, 17, 18]. A structured-output classifier is presented with examples from a joint input-output feature space and it learns to associate inputs with proper outputs by learning a *compatibility function*. Structured-output methods have been applied to a variety of hierarchical classification problems, including applications in natural language processing [18, 14] and prediction of disulfide connectivity [16]. The empirical results in the literature demonstrate that incorporating the structure of the output space into learning often leads to better performance over local learning via binary classifiers.

In this paper we follow the structured-output classification paradigm. Rather than learning a separate classifier for every node in the ontology, we solve the problem by learning a linear predictor in the joint input-output feature space. More specifically, we focus on the structured-perceptron [4] and use it as an alternative to the BLAST nearest-neighbor methodology. Our empirical results demonstrate that learning the structure of the output space yields improved performance over transfer of annotation when both are given the same input-space information (BLAST hits).

A well-known issue in the structured-output approach is the need to consider a potentially exponential number of outputs during inference, and hierarchical classification is no exception. We propose several ways for limiting the size of the search space, and find that this not only leads to efficient inference and training, but also improves classifier accuracy. We also propose a generalization of the F_1 loss function [18] to arbitrary output spaces through the use of kernels, and a variant of the perceptron update rule that leverages the loss function to assess the necessary amount of update. We demonstrate empirically that the modified rule leads to improved accuracy.

2 Methods

Prediction of protein function can be formulated as a hierarchical multi-label classification problem as follows. Each protein is annotated with a macro-label $\mathbf{y} = (y_1, y_2, ..., y_k) \in \{0, 1\}^k$, where each micro-label y_i corresponds to one of the k nodes that belong to the hierarchy defined by the Gene Ontology. The micro-labels take on the value of 1 when the protein performs the function defined by the corresponding node. We refer to such nodes as *positive*. Whenever a protein is associated with a particular micro-label, we also associate it with all its ancestors in the hierarchy, i.e. given a specific term, we associate with it all terms that generalize it. This enforces the constraint that parents of positive nodes are also positive. Throughout this paper we will refer to macro-labels as *output labels* or simply *labels*. We will use the term *entries* when referring to micro-labels. Note that the Gene Ontology consists of three distinct hierarchies: molecular function, biological process and cellular component. In this work we focus on the molecular function hierarchy.

2.1 Measuring performance

Traditionally, classifier performance is measured using the 0-1 loss, which is 0 if the predicted label matches the true labels and 1 otherwise. The average of the loss is used to measure classifier error. In the context of hierarchical classification, the 0-1 loss is not appropriate as it makes no distinction between slight and gross misclassifications. For instance, a label where the protein function is misannotated with its parent or sibling is a better prediction than an annotation in an entirely different subtree. Yet, both will be assigned the same loss since they don't match the true label.

A number of loss functions that incorporate taxonomical information have been proposed in the context of hierarchical classification [8, 14, 9]. These either measure the distance between output labels by finding their least common ancestor in the taxonomy tree [8] or penalize the first inconsistency between the labels in a top-down traversal of the taxonomy [14]. Kiritchenko *et al.* proposed a loss function that is related to the F_1 measure which is used in information retrieval [20] and was used by Tsochantaridis *et al.* in the context of parse tree inference [18]. In what follows we present the F_1 loss function and show how it can be expressed in terms of kernel functions, thereby generalizing it to arbitrary output spaces. The F_1 measure is a combination of precision and recall, which for two-class classification problems are defined as

$$F_1 = \frac{2 \cdot P \cdot R}{P + R},$$

$$P = \frac{tp}{tp + fn}, \quad R = \frac{tp}{tp + fp},$$

where tp is the number of true positives, fn is the number of false negatives and fp is the number of false positives. Rather than expressing precision and recall over the whole set of examples, we express it relative to a single example (known as micro-averaging in information retrieval), computing the precision and recall with respect to the set of micro-labels. Given a vector of true labels ($\mathbf{y}$) and predicted labels ($\hat{\mathbf{y}}$) the number of true positives is the number of micro-labels common to both labels which is given by $\mathbf{y}^T\hat{\mathbf{y}}$. It is easy to verify that

$$P(\mathbf{y}, \hat{\mathbf{y}}) = \frac{\mathbf{y}^T\hat{\mathbf{y}}}{\hat{\mathbf{y}}^T\hat{\mathbf{y}}}, \quad R(\mathbf{y}, \hat{\mathbf{y}}) = \frac{\mathbf{y}^T\hat{\mathbf{y}}}{\mathbf{y}^T\mathbf{y}}. \tag{1}$$

We can now express $F_1(\mathbf{y}, \hat{\mathbf{y}})$ as

$$F_1(\mathbf{y}, \hat{\mathbf{y}}) = \frac{\mathbf{y}^T \hat{\mathbf{y}}}{\mathbf{y}^T \mathbf{y} + \hat{\mathbf{y}}^T \hat{\mathbf{y}}},$$

and define the F_1-loss as $\Delta_{F_1}(\mathbf{y}, \hat{\mathbf{y}}) = (1 - F_1(\mathbf{y}, \hat{\mathbf{y}}))$ [18].

We propose to generalize this loss to arbitrary output spaces by making use of kernels. Replacing dot products with kernels we obtain

$$P(\mathbf{y}, \hat{\mathbf{y}}) = \frac{K(\mathbf{y}, \hat{\mathbf{y}})}{K(\hat{\mathbf{y}}, \hat{\mathbf{y}})} \qquad R(\mathbf{y}, \hat{\mathbf{y}}) = \frac{K(\mathbf{y}, \hat{\mathbf{y}})}{K(\mathbf{y}, \mathbf{y})}.$$

In hierarchical classification, where we use a linear kernel, these definitions yield values identical to those in Equation (1). Expressing precision and recall using kernels leads to the following generalization of the F_1-loss, which we call the *kernel loss*:

$$\Delta_{ker}(\mathbf{y}, \hat{\mathbf{y}}) = 1 - F_1(\mathbf{y}, \hat{\mathbf{y}}) = 1 - \frac{2K(\mathbf{y}, \hat{\mathbf{y}})}{K(\mathbf{y}, \mathbf{y}) + K(\hat{\mathbf{y}}, \hat{\mathbf{y}})}. \tag{2}$$

We used the kernel loss to measure accuracy in our experiments.

2.2 The Structured-Output Perceptron

A standard approach in learning classifiers for two-class classification problems is to learn a discriminant function $f(\mathbf{x})$ and classify the input $\mathbf{x}$ according to the sign of $f(\mathbf{x})$. In structured output learning the discriminant function becomes a function $f(\mathbf{x}, \mathbf{y})$ of both inputs and labels, and can be thought of as measuring the compatibility of the input $\mathbf{x}$ with the output $\mathbf{y}$. We denote by $\mathcal{X}$ the space used to represent our inputs (proteins) and by $\mathcal{Y}$ the set of labels we are willing to consider, which is a subset of $\{0,1\}^k$ for hierarchical multi-label classification. Given an input $\mathbf{x}$ in the input feature space $\mathcal{X}$, structured-output methods infer a label according to:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y} \in \mathcal{Y}} f(\mathbf{x}, \mathbf{y} | \mathbf{w}), \tag{3}$$

where the function : $\mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is parameterized by a vector of parameters $\mathbf{w}$. This classification rule chooses the label $\mathbf{y}$ that is most compatible with an input $\mathbf{x}$. We assume the function is linear in $\mathbf{w}$, i.e. $f(\mathbf{x}, \mathbf{y}|\mathbf{w}) = \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$ in some space defined by the mapping ϕ. Whereas in two class-classification problems the mapping ϕ depends only on the input, in the structured-output setting it is a joint function of inputs and outputs.

We train the classifier using a variant of the perceptron algorithm generalized for structured outputs [4]. Given a set of n training examples $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$, the algorithm attempts to find the vector $\mathbf{w}$ such that the compatibility function values for the correct output and the best runner-up are separated by a user-defined margin γ:

$$\mathbf{w}^T \phi(\mathbf{x}_i, \mathbf{y}_i) - \max_{\mathbf{y} \in \mathcal{Y} \setminus \mathbf{y}_i} \mathbf{w}^T \phi(\mathbf{x}_i, \mathbf{y}) > \gamma \quad \forall i.$$

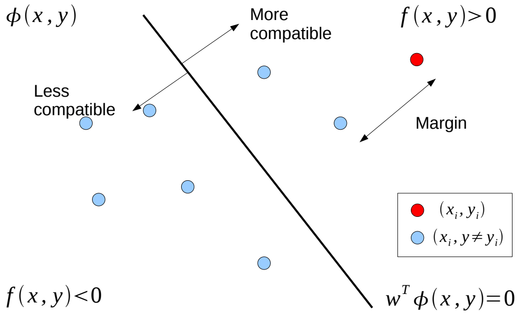


Fig. 1. The geometric view of structured-output classification. Given an input $\mathbf{x}_i$, each point represents input-output label pairs $(\mathbf{x}_i, \mathbf{y})$. This figure represents the ideal case: The corect label has the highest compatibility function value and the second best candidate is separated by a margin. The perceptron defines a linear function over the joint input-output space defined by $\phi(\mathbf{x}, \mathbf{y})$. Unlike the case of two-class classification the hyperplane $f(\mathbf{x}, \mathbf{y}) = 0$ has no special meaning. Correct and incorrect labels may have both positive and negative values.

The geometric intuition behind structured perceptron is presented in Figure 1. Ideally, we would like to learn $\mathbf{w}$ such that the true label has the largest compatibility function value and the second best candidate is separated by the margin γ.

To make use of kernels, we assume that the weight vector $\mathbf{w}$ can be expressed as a linear combination of the training examples:

$$\mathbf{w} = \sum_{j=1}^{n} \sum_{\mathbf{y}' \in \mathcal{Y}} \alpha_{j,\mathbf{y}'} \phi(\mathbf{x}_j, \mathbf{y}').$$

This leads to reparameterization of the compatibility function in terms of the coefficients α:

$$f(\mathbf{x}, \mathbf{y}|\alpha) = \sum_{j=1}^{n} \sum_{\mathbf{y}' \in \mathcal{Y}} \alpha_{j,\mathbf{y}'} K((\mathbf{x}_j, \mathbf{y}'), (\mathbf{x}, \mathbf{y})),$$

where $K : (\mathcal{X} \times \mathcal{Y}) \times (\mathcal{X} \times \mathcal{Y}) \rightarrow \mathbb{R}$ is the joint kernel defined over the input-output space. In this work, we take the joint kernel to be the product of the input space and the output space kernels: $K((\mathbf{x}, \mathbf{y}), (\mathbf{x}', \mathbf{y}')) = K_{\mathcal{X}}(\mathbf{x}, \mathbf{x}') K_{\mathcal{Y}}(\mathbf{y}, \mathbf{y}')$. Our intuition for using a product kernel is that two examples are similar in the input-output feature space if they are similar in both the input and the output spaces. For the output-space kernel, $K_{\mathcal{Y}}$, we use a linear kernel; the input-space

Algorithm 1 Structured Outputs Perceptron

```
Input: training data $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$
Output: parameters $\alpha_{i,\mathbf{y}}$ for $i = 1, ..., n$ and $\mathbf{y} \in \mathcal{Y}$.
Initialize: $\alpha_{i,\mathbf{y}} = 0 \quad \forall i, \mathbf{y}$. //only non-zero values of $\alpha$ are stored explicitly
and the rest are assumed to be 0
repeat
  for $i = 1$ to $n$ do
    Compute the top scoring label that differs from $y_i$:
    $\hat{\mathbf{y}} \leftarrow \arg\max_{\mathbf{y} \in \mathcal{Y} \setminus y_i} f(\mathbf{x}_i, \mathbf{y}|\alpha)$
    Compute the difference in the compatibility function values:
    $\delta = f(\mathbf{x}_i, \mathbf{y}_i) - f(\mathbf{x}_i, \hat{\mathbf{y}})$
    if $\delta < \gamma$ then
      $\alpha_{i,\mathbf{y}_i} \leftarrow \alpha_{i,\mathbf{y}_i} + 1$
      $\alpha_{i,\hat{\mathbf{y}}} \leftarrow \alpha_{i,\hat{\mathbf{y}}} - 1$
    end if
  end for
until a termination criterion is met
```

kernel is described below. In our experiments we normalize the kernel into a cosine-like kernel

$$K_{cosine}((\mathbf{x}, \mathbf{y}), (\mathbf{x}', \mathbf{y}')) = \frac{K((\mathbf{x}, \mathbf{y}), (\mathbf{x}', \mathbf{y}'))}{\sqrt{K((\mathbf{x}, \mathbf{y}), (\mathbf{x}, \mathbf{y}))K((\mathbf{x}', \mathbf{y}'), (\mathbf{x}', \mathbf{y}'))}}$$

as it increased classifier performance.

The perceptron method for learning the coefficients α is presented in Algorithm 1. We start off by finding the label $\hat{\mathbf{y}}$ that yields the highest compatibility function value and that differs from the true label $\mathbf{y}_i$. We then compute δ, the difference in the compatibility function values between the two. A negative value for δ indicates misclassification while a positive value for δ that is smaller than γ indicates a margin violation. In our application, the termination criterion is taken to be a limit on the number of iterations. The α update rules are according to the literature standard [4, 12]. We refer to this version of perceptron as $prcp_{1/-1}$.

The $prcp_{1/-1}$ update adds a constant -1 in the case of a misclassification or margin violation, regardless of whether the classifier made a big mistake or a slight one. Intuitively, we would like to penalize gross misclassifications with larger values. We propose to update the coefficient associated with $\hat{\mathbf{y}}$ by the amount of dissimilarity it has with the true label. This can be done by utilizing the loss function:

$$\alpha_{i,\hat{\mathbf{y}}} \leftarrow \alpha_{i,\hat{\mathbf{y}}} - \Delta_{ker}(\mathbf{y}_i, \hat{\mathbf{y}})$$

Note that the loss is between 0 and 1. Thus, when there is no similarity between the predicted and the true label, the corresponding α coefficient will be updated by -1, as before. Less penalty will be assigned for predicting labels that are more similar to the true label. We refer to the modified version of the perceptron as $prcp_{1/-\Delta}$.

2.3 Inference

The arg max in Equation (3) must be computed over the space of all possible output labels $\mathcal{Y}$. In the context of protein function prediction, this is all possible combinations of functions defined by a few thousand GO terms. Explicitly enumerating all of them is not practical due to the exponential complexity. Fortunately, a protein has only a limited number of functions. Incorporating such a limit reduces the number to be polynomial in the number of GO terms. We further reduce this number in several ways.

During training we limited this space to only those labels that appear in the training dataset. We call this space $\mathcal{Y}_1$ and argue that it makes sense to focus on learning only those labels for which we have training data available.

For inference of test example labels we considered three different output spaces, $\mathcal{Y}_1, \mathcal{Y}_2, \mathcal{Y}_3$, in order to examine the effect of the size of the search space on prediction accuracy. We define $\mathcal{Y}_3(\mathbf{x})$ to be the set of macro-labels that appear in the significant BLAST hits of protein $\mathbf{x}$ (e-values below 10^{-6}). Additionally, $\mathcal{Y}_2(\mathbf{x})$ is obtained by taking all the leaf nodes represented in $\mathcal{Y}_3(\mathbf{x})$ and considering all macro-labels consisting of three leaf nodes at the most. Note that the label spaces satisfy: $\mathcal{Y}_3(\mathbf{x}) \subseteq \mathcal{Y}_2(\mathbf{x}) \subseteq \mathcal{Y}_1$.

3 Data Preparation and Experimental Setup

We used the data from the following four species: *C. elegans*, *D. melanogaster*, *S. cerevisiae* and *S. pombe*. Sequence data was obtained from the genome database of each organism (http://www.wormbase.org/, http://flybase.bio.indiana.edu/, http://www.yeastgenome.org/) and annotations were obtained from the Gene Ontology website at http://www.geneontology.org. Our experiments followed the leave-one-species-out paradigm [21], where we withheld one species for testing and trained the perceptron on the remaining data, rotating the species that got withheld. This variant of cross-validation simulates the situation of annotating a newly-sequenced genome. In developing our methods we used the GO-slims ontology; to avoid overfitting we then report results on the full GO ontology. In our analysis we considered all GO molecular function terms that appear as annotations in at least 10 proteins, resulting in a total of 361 nodes.

To prepare the data we removed all annotations that were discovered through computational means as these are generally inferred from sequence or structure similarity and would introduce bias into any classifier that used sequence similarity to make a prediction [13]. This was done by removing all annotations with the evidence codes: IEA, ISS, ND, RCA, and NR. Note that limiting the experiments to annotations that were possibly derived by computational means limits the number of species that can be considered to a very small set of model organisms, and for simplicity we focused on the eukaryotes listed above.

We then ran BLAST for each of the proteins in our dataset against all four species, removing the hits where the protein was aligned to itself. We employed the nearest neighbor BLAST methodology as our baseline. For every test protein, we transferred the annotations from the most significant BLAST hit against a protein from another species. Proteins which didn't have a hit with an e-value below 10^{-6} were not considered in our experiments.

The structured-output perceptron is provided exactly the same data as the BLAST method. The input-space kernel is an empirical kernel map [15] that uses the negative-log of the BLAST e-values that are below 50, where the features were normalized to have values less than 1.0. An empirical kernel map arises from the intuition that two similar proteins will have similar patterns of similarity to proteins in the database, i.e. their vectors of e-values will be similar.

We ran five fold cross-validation on the training data to select a suitable value of the margin parameter γ for each left-out species. In our experiments, we noticed that finding the right value of γ is not as essential as using the loss update proposed in the previous section.

4 Results

Test on # proteins	*C. elegans* 844	*D. melanogaster* 1804	*S. cerevisiae* 1853	*S. pombe* 898	Output Space
BLAST NN	0.523	0.379	0.354	0.329	
$prcp_{1/-1}$	0.542	0.399	0.384	0.363	$\mathcal{Y}_1$
$prcp_{1/-1}$	0.540	0.383	0.319	0.341	$\mathcal{Y}_2$
$prcp_{1/-1}$	0.527	0.347	**0.314**	0.312	$\mathcal{Y}_3$
$prcp_{1/-\Delta}$	**0.500**	0.346	0.366	0.330	$\mathcal{Y}_1$
$prcp_{1/-\Delta}$	0.521	0.352	0.322	0.312	$\mathcal{Y}_2$
$prcp_{1/-\Delta}$	0.508	**0.331**	0.322	**0.295**	$\mathcal{Y}_3$
Random	0.660	0.730	0.730	0.704	

Table 1. Classification results on predicting GO molecular function terms (361 terms that have more than 10 annotations). We compare traditional transfer-of-annotation (BLAST NN) with two variants of the perceptron across three methods for limiting the output space. Reported is mean kernel loss per protein for each algorithm. The number of proteins used in each organism is displayed in the second row. For comparison, we also include the performance of a random classifier that transfers annotation from a training example chosen uniformly at random. The standard deviation estimated for the presented performance was between 0.003 and 0.01.

The results for the leave-one-species-out experiments are presented in Table 1. The results show that the structured perceptron outperforms the BLAST nearest-neighbor classifier. Before looking at the differences between the two variants of the perceptron and the different sets of output labels considered, we note that all the classifiers performed poorly on *C. elegans*. This is due to the fact that a vast majority of proteins in this species are annotated as protein binders

(GOID:0005515). Such annotations contain little information from a biological standpoint and result in a skewed set of output labels.

Our results show that except for one case (*C. cerevisiae*), the $prcp_{1/-\Delta}$ method which uses the loss function in the update rule of the perceptron outperformed the standard $prcp_{1/-1}$. Furthermore, excluding *C. elegans*, using the restricted output spaces $\mathcal{Y}_2$ or $\mathcal{Y}_3$ resulted in better performance than using the full output space $\mathcal{Y}_1$, with best performance being obtained using $\mathcal{Y}_3$ which is the most restricted output space. The larger label-space $\mathcal{Y}_1$, results in the inference procedure considering many annotations that are irrelevant to the actual function of the protein, which can reduce prediction accuracy. When used in conjunction with $\mathcal{Y}_2$ or $\mathcal{Y}_3$ our structured-outputs method can be thought of as prioritizing the annotations suggested by BLAST in a way that uses the structure of the Gene Ontology hierarchy. The results support our hypothesis that learning the structure of the output space is superior to performing transfer of annotations.

To assess the robustness of our classifier we ran an additional experiment where 20% of the training data was chosen at random and withheld from the training. The classifier was then trained on the remaining 80% of the training data and tested as before. This provided us with a standard deviation measure that indicated how consistent the classifiers were at obtaining the performance presented in Table 1. We computed the standard deviations across 30 trials for every classifier. The values for BLAST nearest-neighbors and the random classifier were in the range (0.004, 0.009). The structured-output perceptron with a $1/-1$ update had standard deviation values in the range (0.006, 0.010), while the structured output-perceptron with a $1/-\Delta$ update yielded more consistent performance with the standard deviation values in the range (0.003, 0.007).

5 Conclusions

We have shown that a structured output method performs better than a nearest neighbor method when provided with the same information. Our structured output method can be enhanced in several ways to further boost its performance: Additional information can easily be provided in the form of additional kernels on the input space that use other forms of genomic information (e.g. protein-protein interactions); the structured-perceptron can be replaced with maximum margin classifiers [18, 14]; and furthermore, semi-supervised learning can be used to leverage the abundance of available sequence information. In future work we will also consider larger datasets that include a larger number of species.

References

1. S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. Basic local alignment search tool. *J. Mol. Biol*, 215(3):403–410, 1990.
2. Y. Altun, I. Tsochantaridis, and T. Hofmann. Hidden markov support vector machines. *Proc. ICML*, 6, 2003.

3. Z. Barutcuoglu, R.E. Schapire, and O.G. Troyanskaya. Hierarchical multi-label prediction of gene function. *Bioinformatics*, 22(7):830–836, 2006.
4. M. Collins. Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms. *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 1–8, 2002.
5. M. Deng, T. Chen, and F. Sun. An integrated probabilistic model for functional prediction of proteins. In *RECOMB*, pages 95–103, 2003.
6. Michael Y. Galperin and Eugene V. Koonin. Sources of systematic error in functional annotation of genomes: Domain rearrangement, non-orthologous gene displacement and operon disruption. *In Silico Biology*, 1(1):55–67, 1998.
7. Gene Ontology Consortium. Gene ontology: tool for the unification of biology. *Nat. Genet.*, 25(1):25–9, 2000.
8. T. Hofmann, L. Cai, and M. Ciaramita. Learning with taxonomies: Classifying documents and words. *NIPS Workshop on Syntax, Semantics, and Statistics*, 2003.
9. Svetlana Kiritchenko, Stan Matwin, and A. Fazel Famili. Functional annotation of genes using hierarchical text categorization. In *Proc. of the BioLINK SIG: Linking Literature, Information and Knowledge for Biology, a joint meeting of the ISMB BioLINK Special Interest Group on Text Data Mining and the ACL Workshop on Linking Biological Literature, Ontologies and Databases*, 2005.
10. G. R. G. Lanckriet, T. De Bie, N. Cristianini, M. I. Jordan, and W. S. Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20(16):2626–2635, 2004.
11. D. Pal and D. Eisenberg. Inference of protein function from protein structure. *Structure*, 13:121–130, January 2005.
12. V. Punyakanok, D. Roth, W. Yih, and D. Zimak. Learning and inference over constrained output. *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1124–1129, 2005.
13. M. Rogers and A. Ben-Hur. Assessment bias in predicting protein function. in preparation.
14. J. Rousu, C. Saunders, S. Szedmak, and J. Shawe-Taylor. Kernel-Based Learning of Hierarchical Multilabel Classification Models. *The Journal of Machine Learning Research*, 7:1601–1626, 2006.
15. B. Scholkopf, J. Weston, E. Eskin, C. Leslie, and W.S. Noble. A kernel approach for learning from almost orthogonal patterns. *Proceedings of the 13th European Conference on Machine Learning*, pages 511–528, 2002.
16. B. Taskar, V. Chatalbashev, D. Koller, and C. Guestrin. Learning structured prediction models: A large margin approach. In *Twenty Second International Conference on Machine Learning (ICML05)*, 2005.
17. B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. *Advances in Neural Information Processing Systems*, 16:51, 2004.
18. I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large Margin Methods for Structured and Interdependent Output Variables. *The Journal of Machine Learning Research*, 6:1453–1484, 2005.
19. K. Tsuda, H.J. Shin, and B. Schölkopf. Fast protein classification with multiple networks. In *ECCB*, 2005.
20. CJ Van Rijsbergen. *Information Retrieval.* Butterworth-Heinemann Newton, MA, USA, 1979.
21. A. Vinayagam, R. K onig, J. Moormann, F. Schubert, R. Eils, K.-H. Glatting, and S. Suhai. Applying support vector machines for gene ontology based gene function prediction. *BMC Bioinformatics*, 5:178, 2004.

GOPred: Combining classifiers on the GO

Ömer Sinan Saraç[1], Rengül Çetin-Atalay[2], and Volkan Atalay[1]

[1] Department of Computer Engineering, Middle East Technical University, Ankara TURKEY,
sarac,volkan@ceng.metu.edu.tr
[2] Department of Molecular Biology and Genetics, Bilkent University, Ankara TURKEY
rengul@bilkent.edu.tr

Abstract. Functional annotation of proteins is an important problem in computational biology. There is a wide range of methods developed in the literature using features such as motifs, domains, homology, structure and physicochemical properties. Since information obtained using any of these features depends on the function to be assigned to the protein, there is no single method that performs best in all functional classification problems. In this study, we investigated the effect of combining different methods to form a more accurate classifier. First, we formulated the function annotation problem as a classification problem defined on 300 different Gene Ontology (GO) terms from Molecular Function aspect. We present a method to form positive and negative training examples while taking into account the Directed Acyclic Graph structure and evidence codes of GO. We applied 3 different methods and their combination to this classification problem. Results show that combining different methods improves prediction accuracy in most of the cases. The proposed method, GOPred, is available as an online annotation tool.
Availability: http://kinaz.fen.bilkent.edu.tr/gopred

1 Introduction

Attempts to automated function annotation follow two main tracks in the literature. In the first track, the target protein to be annotated is searched against public databases of already annotated proteins. Annotations of the highest scoring hits, according to some similarity calculation, are considered to be transfered onto the target protein. We call this track the *transfer approach.* Despite some known drawbacks such as, excessive transfer of annotations, low sensitivity/specificity, propagation of database errors, this track is the most widely used among the biologists ([24, 13]).

In the second track, annotation of proteins is formulated as a classification problem where the annotations are classes and proteins are samples to be classified. This *classification approach* allows scientists to use sophisticated and powerful classification algorithms such as support vector machines (SVM) and artificial neural networks (ANN). These methods explicitly form a boundary between the negative and positive training samples and are shown to be more accurate in many cases ([20]). Yet, they are not as popular among biologist as one would expect. One reason is that, classification approach requires well defined classes and positive and negative training data for each class. But protein function is a vague term where the exact meaning depends on the context in which it is used ([13]). Data preparation is not straightforward since functional terms are related to each other and proteins may have more than one annotation. We believe that if one can establish a classification framework with rich number of important functional terms and high quality training data, methods in classification approach will receive more attention.

There is a wide range of classification approaches to automated functional annotation in the literature. They can be grouped into three categories depending on the employed features:

1. homology-based approaches,
2. subsequence-based approaches,
3. feature-based approaches.

Homology-based approaches utilize overall sequence similarity of the target protein to the positive and negative training data to decide which functional class it belongs. Most well-known and widely used methods for finding sequence similarity is local alignment search tools such as BLAST and PSI-BLAST ([1, 2]). Subsequence-based approaches focus on highly conserved subregions such as motifs or domains that are critical for a protein to perform a specific function. These methods are especially effective when function to be assigned requires a specific motif or domain. Existence of these highly conserved regions in a protein enables us to infer a specific annotation even in remote homology situations ([6, 19, 23]. In the feature-based approach, biologically meaningful properties of a protein such as frequency of residues, molecular weight, secondary structure, extinction coefficients are extracted from the primary sequence. These properties are then arranged as feature vectors and used as input to classification techniques such as artificial neural networks (ANN) or support vector machines (SVM) ([16, 8, 17, 9]). Each of these approaches may have different strengths and weaknesses on the classification of different functional terms. As a result, combining methods from different approaches may be more successful on classification of a wide range of protein functions.

In this study, we developed a method to prepare training data for the terms defined in Gene Ontology (GO) framework ([4]). We focused on annotation of proteins with 300 GO molecular function terms where we formulated this problem as a classification task with 300 classes. We applied 3 different classification methods. In a one-versus-all setting, usually the size of negative training dataset is much larger than that of the positive training dataset. In order to avoid a bias towards larger negative class, we present a threshold relaxation method that not only shifts the threshold towards the more appropriate classification boundary but also maps the output of the classifier to a probability that states how probable it is that the given sample is a member of the target classes. Finally, we investigated different classifier combination methods and results showed that combination improved the performance for about 93% of the classifiers while yielding similar results to the best performing method for the rest of the classifiers.

2 Dataset

One of the well-known and most widely used attempt to standardize protein function terms and to define their relations is Gene Ontology (GO). GO provides ontology in 3 aspects: *molecular function*, *biological process*, and *cellular location*. In this study, we focus on *molecular function* aspect. GO organizes molecular functions as nodes on a directed acyclic graph (DAG). Each node is a more specific case of its parent node or nodes. Here, we present a way of establishing positive and negative training data for each class by using evidence codes provided by the GO Annotation (GOA) project and by considering the structure of the GO DAG. While preparing training data, we used Uniprot release 13.0 as the source for protein sequences([5]). Annotations are obtained from October, 2007 version of GOA mapping file and again October 2007 version of GO ontology is used as the basis of our functional terms and their relations in our system.

Preparing positive training dataset is relatively easy compared to negatives. First we extracted all proteins that are annotated with the target term or one of its descendants connected with a *is_a* relation by the Gene Ontology Annotation (GOA) project. In order to populate a training dataset without any bias towards computational prediction methods and to reduce the noise in the training data as much as possible, we filtered out those proteins that are annotated with one of IC, IEA, ISS, NAS, ND evidence codes. These codes refer to annotations either obtained by electronic means or have ambiguity in their origin ([11]). The rest of the evidence codes IDA, IEP, IGI, IMP, IPI, RCA, and TAS refer to experimental evidences which we think are more reliable.

Theoretically, an annotation for a protein only specifies what function it performs. This is not (generally) an indication of what it doesn't perform. For a protein not having a specific functional label might be merely due to lack of knowledge or experiment. Although this may not be a severe problem in practice, it helps us to understand the difficulties of constructing a negative training dataset for a target term. As a result, each protein that does not have the annotation of the target class or one of its descendants is a possible negative training sample. Including all such proteins in the negative training dataset is neither useful nor necessary. First of all, sizes of the positive and negative training sets may become very unbalanced in such a case. For some functional classes, the size of positive training dataset is on the order of tens of proteins, whereas it is about tens of thousands for the negative dataset. Second, computational cost increases with the size of the negative training dataset.

Since we trained our classifiers in one-versus-all setting for 300 GO molecular function terms, our strategy was to select random representative sequences (at most 10) from each term other than the target term. We imposed two constraints on the selected random representative sequences:

1. A sequence shouldn't be annotated with the target term or one of it is descendant terms.
2. If a sequence is annotated with one of the ancestors of the target term, it should also have been annotated with a sibling of the target term.

The first constraint is trivial since we don't want to include protein sequences that are already in the positive training data. Second constraint is imposed in order to avoid including prospective positive training data in to the negative dataset. Ideally, each protein should be annotated with a GO term on a leaf node, in other words, with most specific annotation. If a protein is annotated only up to an internal node, this means either there is lack of evidence for a more specific annotation or an appropriate GO term for that protein has not been added to the ontology yet. Thus, we excluded proteins that are annotated by an ancestor GO term but not with a sibling.

3 Methods

After preparing positive and negative training data for each of 300 GO molecular function terms, we applied three classification methods representing three approaches:

- BLAST k-nearest neighbor (BLAST-kNN) for homology-based approach,
- Subsequence Profile Map (SPMap) for subsequence-based approach,
- Peptide statistics combined with SVMs (PEPSTATS-SVM) for feature-based approach.

3.1 BLAST-kNN

In order to classify the target protein, we used k-nearest neighbor algorithm ([Cover and Hart, 1967]). Similarities between the target protein and proteins in the training data were calculated using NCBI-BLAST tool. We extracted k-nearest neighbors having the highest k BLAST score. The output of BLAST-kNN, O_B for a target protein is calculated as:

$$O_B = \frac{S_p - S_n}{S_p + S_n} \tag{1}$$

where S_p is the sum of BLAST scores of proteins in k-nearest neighbors that are in the positive training data. Similarly, S_n is the sum of scores of k-nearest neighbor proteins that are in the negative training data. Note that the value of O_B is between -1 and +1. The output is 1 if all k nearest proteins are the elements of positive training dataset and -1 if all k proteins are from negative training dataset. Instead of directly using O_B with a fixed threshold we used the threshold relaxation algorithm given in Section 3.4.

3.2 SPMap

SPMap maps protein sequences to a fixed-dimensional feature vector where each dimension represents a group of similar fixed-length subsequences. In order to obtain groups of similar subsequences, SPMap first extracts all possible subsequences from the positive training data and clusters similar subsequences. A probabilistic profile or a position specific scoring matrix is then generated for each cluster. The number of clusters determine the dimension of the feature space. Generation of these profiles is called the construction of the feature space map. Once this map is constructed, it is used to represent protein sequences as fixed dimensional vectors. Each dimension of the feature vector is the probability calculated by the best matching subsequence of the protein sequence to the corresponding probabilistic profile. If the sequence to be mapped contains a subsequence similar to a specific group, the value of the corresponding dimension will be high. Note that this representation reflects the information of subsequences that are highly conserved among the positive training data. After the construction of the feature vectors, SVMs are used as to train classifiers. Further information on SPMap is found in [23].

3.3 PEPSTATS-SVM

Pepstats tool which is a part of the European Molecular Biology Open Software Suite (EMBOSS) is used to extract peptide statistics of the proteins ([22]). Each protein is represented by a 37 dimensional vector. Peptide features include *molecular weight*, *charge*, *isoelectric point*, *Dayhoff statistics for each aminocid*, *extinction coefficients*, *percentage of residue groups*, etc. These features are scaled using the ranges of positive training data and finally fed to an SVM classifier.

3.4 Threshold Relaxation

Optimization algorithm of SVM that finds the hyperplane maximizing the margin between the hyperplane and the training data is data-driven and may have bias towards the classes with more training samples. As a result, using the natural threshold 0 usually results in poor sensitivity if the sizes of the positive and negative training datasets are unbalanced. This is exactly the case in our problem. There are studies in the literature about threshold relaxation towards smaller class ([3, 26]). In our study, instead of adjusting the threshold value, we present a method that defines probability $P(x)$ of a sample x to be in the positive class.

First, we split the test data into two sets, a *helper set*, to calculate the probability $P(x)$ and a held-out *validation set*, to evaluate the performance of the method. Since, the number of positive test samples is outnumbered by the negative test samples, our method should handle this unbalanced situation. Thus, we calculated a confidence value for the new sample for being positive and negative separately and then we combined these confidences into a single probability. The confidence for a new sample being positive $C_p(x)$ is calculated as the ratio of positive samples in helper set having a classifier output lower than that of the new sample. The confidence for being negative $C_n(x)$ is calculated similarly (Equation 2 and Equation 3). These ratios are combined to calculate the probability of the new sample to be in positive class (Equation 4). A new sample is predicted as positive if $C > 0.5$ and as negative, otherwise.

$$C_p(x) = \frac{\sum_{y \epsilon Y_p} I(\phi(x) >= \phi(y))}{|Y_p|} \tag{2}$$

$$C_n(x) = \frac{\sum_{y \epsilon Y_n} I(\phi(x) <= \phi(y))}{|Y_n|} \tag{3}$$

$$P(x) = \frac{C_p}{C_p + C_n} \tag{4}$$

Y_p and Y_n are the positive and negative test samples in the helper set, respectively. $\phi(x)$ denotes the output of the classifier for sample x. I operator returns 1 if the condition holds, 0 otherwise. Note that this method implicitly adjusts the threshold. Furthermore, it provides the user a measure to assess how probable it is that the sample is a member of the given class.

3.5 Classifier Combination

Observations on many classification problems with different classification methods have shown that although there is usually a best performing method on a specific problem, the samples that are correctly classified or misclassified by different methods may not necessarily overlap ([18]). This observation led to the idea of classifier combination in order to achieve a higher accuracy ([18, 27]). In this study we investigated four classifier combination techniques for three different classification methods each one representing one of the three approaches stated in Section 1; **Voting, Mean, Weighted Mean** and **Addition**.

Voting, also known as majority voting, simply decides the class of the new sample by counting positive and negative votes from each classifier. Note that votes of the methods have equal weight and the output value of the classifiers are not taken into account.

For the *Mean* combination method, the mean of the probability values calculated by Equation 4 is used to decide the class of the new sample. If this mean value is greater than 0.5 sample is labeled as positive.

The combination method *Mean* treats each method equally. But the performance of the methods vary for different functional classes. Thus in the *weighted mean* method, we assigned weights to each method depending on their classification performance on the functional class for which the classifier combination is performed. To assess the performance of the methods we made use of the area under the Receiver Operating Characteristic (ROC) curve, which is called the ROC score. ROC score is a widely used as measure to evaluate the performance of classification methods. ROC score gives an estimation of the discriminative power of the method independent of the threshold value. To calculate the ROC score of each method we used the *helper test set.* Then, we assigned a weight to each method calculated by Equation 5.

$$W(m) = \frac{R_m^4}{\sum_{n \epsilon Methods} R_n^4} \tag{5}$$

$W(m)$ denotes weight of method m. R_m is the ROC score for method m. Note that we used the 4^{th} power of ROC scores to assign higher weight to the method with a better ROC score.

In the *Addition* method, output value of the classification methods are summed directly. The probability defined in Equation 4 is then calculated using these added values.

4 Results and Discussion

Tests were performed for 300 GO terms in one-versus-all setting. For each GO term, statistics are obtained by averaging results from 5-fold cross-validation. In order to calculate the probability described in Equation 4, we used leave-one-out cross validation in the test set. In other words, we used all available test dataset but one as the *helper set* and one held-out sample as the *validation set.* This is performed for all of the test dataset.

In order to compare the methods and combination strategies, we made use of F_1 statistics. When the sizes of the positive and negative test sets are unbalanced several common statistics such as, sensitivity, specificity, and accuracy may overstate or understate the performance of the classification. F_1 measure is the harmonic mean between precision and sensitivity. It is robust in case of uneven datasets ([15]).

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{7}$$

$$F_1 = \frac{2xPrecision \times Sensitivity}{Sensitivity + Precision} = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{8}$$

TP, FP, TN, and FN denotes true positive, false positive, true negative and false negative, respectively.

Weighted mean method performed best in 279 of 300 classifiers, with an average F_1 score of 0.77. Thus, it is chosen to be the basis combination method for our online tool *GOPred. Addition* was the best for 8 classes. *Voting* and *mean* were the best methods for 1 and 3 of the classes, respectively. On the overall, combination improved the performance for 291 of 300 classes. One should note that for the rest of the cases, at least one combination method performed very similar to the best performing single method. Average senstivity, specificity and F_1 scores over 300 classes is given in Table 4. With respect to F_1 scores, BLAST-kNN turned out to be best performing single method for a majority of the functional terms while outperformed by SPMap only at a small fraction of functional terms. Pepstats-SVM was the weakest method in all functional classes. Results show that simple peptide statistics are not sufficient for accurate classification of GO functional terms. Nevertheless, it turned out to be that samples correctly classified by each of the methods do not overlap. This explains the success of the combination methods. As a future work, Pepstats-SVM will be replaced by a more powerful feature-based classification method.

In order to investigate the effect of *threshold relaxation* method presented in Section 3.4 we repeated the whole experiment by using natural threshold 0 for all methods. Figure 1 shows the comparison of sensitivity

Table 1: F_1 scores, sensitivity and specificity values averaged over 300 GO functional term classifiers

Method	F_1	Sensitivity	Specificity
SPMap	0.62	89.12	88.92
BLAST-kNN	0.70	92.07	92.53
Pepstats-SVM	0.39	75.47	75.48
Voting	0.71	90.50	92.85
Mean	0.74	91.11	93.74
Weighted Mean	0.77	91.82	94.79
Addition	0.70	92.72	92.49

and specificity values with and without threshold relaxation averaged over 300 GO terms. Pepstats-SVM turned out to be the most benefiting method which is actually useless without threshold adjustment. BLAST-kNN is the less effected method which is not surprising since k-nearest neighbors method do not generate a single decision boundary. After threshold relaxation there is a small decrease in specificity but a much larger increase in sensitivity. This conforms with our expectation that there will be a bias towards the class with more training samples. Automated function prediction tools are generally used to have a rough idea about the protein's possible functions before conducting further in vitro experiments. We believe that failing to detect an important annotation is a more severe problem than assigning a wrong annotation. Thus, increasing sensitivity without a detrimental effect to specificity is a very important achievement. Detailed statistics (Dataset sizes, TP, FP, TN, FN, Sensitivity, Specificity, ROC score, F_1 score) for all of the methods on each GO functional term can be found in supplementary material.

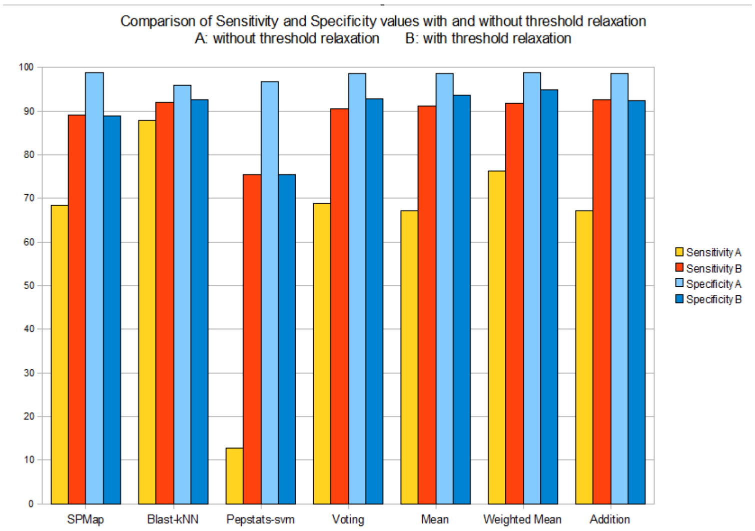


Fig. 1: Comparison of average sensitivity and specificity values with and without threshold relaxation

The actual challenge for an automated annotation tool is the annotation of newly identified sequences or genomes. Thus, we applied our method to the prediction of functions of 8 newly reported Homo Sapiens proteins to NCBI in the last year. The combined classifiers were able to predict the reported functions of the proteins in all of the cases. This is a good indication of the effectiveness of the method. Table 4 shows proteins, their reported functions, and annotations of GOPred along with the probabilities calculated by our method that the protein can be annotated with the corresponding GO term. Furthermore, GOPred is also

Table 2: GOPred annotations for 8 newly validated human gene entries from NCBI gene database.

Gene Symbol	Reported Function	GOPred annotations:Probability
killin	Nuclear inhibitor of DNA synthesis with high affinity DNA binding [10]	Exonuclease activity: **0.95**
glrx1	glutaredoxin-like, oxidoreductase[12]	oxidoreductase activity: **0.97**
fnip2	AMPK and FLCN interaction[14]	enzyme activator activity: **0.61** enzyme binding: **0.71**
kif18b	microtubule associated motor protein which use ATP[29]	microtubule binding: **0.88** motor activity: **0.83** nucleotide binding: **0.91**
helt	transcription regulator activity[25]	protein homodimerization activity: **0.98** transcription corepressor activity: **0.95**
rgl4	guanin nucleotide dissociation[7]	guanyl-nucleotide exchange factor: **0.79** small GTPase binding: **0.73**
pgap1	GPI inositol-deacylase[28]	lipase activity: **0.89** hydrolase activity acting on ester bonds: **0.89** acyltransferase activity: **0.79**
cobra1	member of negative elongation factor complex during transcription, inhibitor of AP1[21]	ribonucleotide binding: **0.91** enzyme regulator activity: **0.81**

applied to annotation of 73 newly reported genes from Ovis Aries (Sheep). Results are available on GOPred web site (http://kinaz.fen.bilkent.edu.tr/gopred/ovisaries.html).

5 Conclusion

Automated functional annotation of proteins is an important and difficult problem in computational biology. Most of the function prediction tools, aside from those that uses simple *transfer* approach, defines the annotation problem as a classification problem. Thus, they require positive and negative training data and the success of the resulting classifier relies on the representative power of this dataset. In this study, we first presented a method to construct accurate positive and negative training data using DAG structure of GO and annotations and evidence codes provided by GOA project.

There is a rich literature on automated function prediction methods each of which have different strengths and weaknesses. We investigated the effects of combining different classifiers for accurate annotation of proteins with functional terms defined in molecular function aspect of GO. Resulting combined classifier clearly outperformed constituent classifiers. Test results also showed that the best combination strategy is the *weighted mean* classifier combination method which assigns different weights to classifiers depending on their discriminative strengths on a specific functional term.

It is also important to note that we do not merely give annotations. We also present a threshold relaxation method that not only avoids the bias towards the class with more training data but also assigns a probability to the prediction which provides a way of assessing the strength of the annotation. This means we also provide less probable functional annotations. This information may help the biologist to build a road map before conducting expensive in vitro experiments.

References

1. Altschul,S.F., Gish,W., Miller,W., Myers,E.W., Lipman,D.J (1990) A basic local alignment search tool, *Journal of Molecular Biology*, **215**, 403-410.
2. Altschul,S.F., Madden,T.L., Schaffer,A.A., Zhang,J., Zhang,Z., Miller,W., Lipman,D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, *Nucleic Acid Research*, **25**, 3389-3402.
3. Arampatzis,A. (2002) Unbiased S-D Threshold Optimization, Initial Query Degradation, Decay, and Incrementality, for Adaptive Document Filtering, *Tenth Text Retrieval Conference* (TREC-2001), 596-605.
4. Ashburner,M., Ball,C., Blake,J., Botstein,D., Butler,H., Cherry,J., Davis,A., Dolinski,K., Dwight,S., Eppig,J. et. al, Gene Ontology: tool for the unification of biology, 2000, Nature Genetics 25, 25-29.

5. Bairoch,A., Apweiler,R., Wu,C.H., Barker,W.C., Boeckmann,B., Ferro,S., Gasteiger,E., Huang,H., Lopez,R., Magrane,M., Martin,M.J., Natale,D.A., O'Donovan,C., Redaschi,N., Yeh,L.S. (2005) The Universal Protein Resource (UniProt). *Nucleic Acids Research.* **33** D154-D159.

6. Ben-hur,A., Brutlab,D. (2003) Remote homology detection: a motif based approach, *Bioinformatics*, **19**, 26-33.

7. Bodemann,B.O., White,M.A. (2008) Ral GTPases and cancer: linchpin support of the tumorigenic platform, *Nat. Rev. Cancer*, **8(2)**, 133-140.

8. Cai,C., Han,L., Ji,Z., Chen,X., and Chen,Y. (2003) SVM-Prot: Web-based support vector machine software for functional classification of a protein from its primary sequence, *Nucleic Acids Research*, **31(13)**, 3692-3697.

9. Cheng,B.Y.M., Carbonell,J.G., and Klein-Seetharaman,J. (2005) Protein classification based on text document classification techniques. *Proteins*, **58(4)**, 955-970.

10. Cho,Y.J., Liang,P. (2008) Killin is a p53-regulated nuclear inhibitor of DNA synthesis, *Proc Natl. Acad. Sci. USA*, **105(14)**, 5396-5401.

[Cover and Hart, 1967] Cover,T.M. and Hart,P.E. (1967) Nearest neighbor pattern classification. *IEEE Trans. IT*, *13(1)*, 21-27.

11. Eisner,R., Poulin,B., Szafron,D., Lu,P., and Greiner,R., (2005) Improving Protein Function Prediction using the Hierarchical Structure of the Gene Ontology, *IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology.*

12. Fernandes,A.P., Holmgren,A. (2004) Glutaredoxins: glutathione-dependent redox enzymes with functions far beyond a simple thioredoxin backup system, *Antioxid Redox Signal*, **6(1)**, 63-74.

13. Friedberg,I. (2006) Automated protein function prediction - the genomic challenge *Briefings in Bioinformatics*, **7**, 225-242.

14. Hasumi,H., Baba,M., Hong,S.B., Hasumi,Y., Huang,Y., Yao,M., Valera,V.A., Linehan,W.M., Schmidt,L.S. (2008) Identification and characterization of a novel folliculin-interacting protein FNIP2, *Gene*, **415(1-2)**, 60-67.

15. Holloway,D.T., Kon,M.A., and DeLisi,C. (2006) Machine Learning Methods for Transcription data Integration, *IBM Journal of Research and Development*, **50(6)**, 631-643.

16. Jensen,L.J., Gupta,R., Blom,N., Devos,D., Tamames,J., Kesmir,C., Nielsen, H., Staerfeldt,H.H., Rapacki,K., Workman,C., Andersen,C.A.F., Knudsen,S., Krogh,A., Valencia,A., and Brunak, S. (2002) Prediction of human protein function from post-translational modifications and localization features, *J Mol Biol.*, **319(5)**, 1257-1265.

17. Karchin,R., Karplus,K., and Haussler,D. (2002) Classifying G-protein coupled receptors with support vector machines, *Bioinformatics*, **18(1)**, 147-159.

18. Kittler,J.,Hatef,M.,Duin,R.P.W, Matas,J. (1998) On Combining Classifiers, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **20(3)**.

19. Kunik,V., Solan,Z., Edelman,S., Ruppin,E., and Horn,D. (2005) Motif extraction and protein classification, *In Proc. Computational Systems Bioinformatics (CSB)*, 80-85.

20. Leslie,C.S., Eskin,E., Cohen,A., Weston,J., Noble,W.S. (2004) Mismatch string kernels for discriminative protein classification, *Bioinformatics*, **20**, 467-476.

21. McChesney,P.A., Aiyar,S.E., Lee,O.J., Zaika,A., Moskaluk,C., Li,R., El-Rifai,W. (2006) Cofactor of BRCA1: a novel transcription factor regulator in upper gastrointestinal adenocarcinomas *Cancer Research*, **66(3)**, 1346-53.

22. Rice,P., Longden,I., and Bleasby,A. (2000) The European Molecular Biology Open Software Suite, *Trends in Genetics*, **16(6)**, 276-277.

23. Sarac,O.S., Yuzugullu,O.G., Cetin-Atalay,R., and Atalay,V. (2008) Subsequence-based feature map for protein function classification, *Computational Biology and Chemistry*, **32**, 122-130.

24. Sasson,O., Kaplan,N., Linial,M. (2006) Functional annotation prediction: All for one and one for all, *Protein Science*, **15**, 1-16.

25. Schwanbeck,R., Schroeder,T., Henning,K., Kohlhof,H., Rieber,N., Erfurth,M.L., Just,U. (2008) Notch Signaling in Embryonic and Adult Myelopoiesis, *Cell Tissues Organs*, Epub ahead of print.

26. Shanahan,J.G., Roma,N., (2003) Boosting support vector machines for text classification through parameter-free threshold relaxation, *Proc. of* 12^th *int. conf. on Information and knowledge management*, LA, USA, 247-254.

27. Sohn,S.Y., Shin,H.W. (2007) Experimental study for the comparison of classifier combination methods, *Pattern Recognition*, **40(1)**, 33-40.

28. Tanaka,S., Maeda,Y., Tashima,Y., Kinoshita,T. (2004) Inositol deacylation of glycosylphosphatidylinositol-anchored proteins is mediated by mammalian PGAP1 and yeast Bst1p, *J. Biol. Chem*, **279(14)**, 14256-63.

29. Yildiz,A., Selvin,P.R. (2005) Kinesin: walking, crawling and sliding along?, *Trends Cell Biol.*, **15(2)**, 112-120.

Modeling Networks as Probabilistic Sequences of Frequent Subgraphs

Koenraad Van Leemput[1] and Alain Verschoren[2]

[1] Advanced Database Research and Modelling (ADReM),
[2] Intelligent Systems Laboratory (ISLab),
University of Antwerp, Middelheimlaan 1, B-2020 Antwerpen, Belgium

Abstract. Graphs or networks are used as a representation of data in many different areas, ranging from Biology to the World Wide Web. In this paper, a novel approach to graph characterization based on a probabilistic (de)composition into a linear sequence of frequent subgraphs is presented. The resulting probabilistic models are generative for a family of graphs sharing common structural properties. An evolutionary computing approach is used to learn the model parameters for unknown graph classes. This paper describes the (de)composition procedure and illustrates its use in characterizing and discriminating a number of graph types. To demonstrate its practical usefulness, the method is applied to the problem of modeling transcriptional regulatory networks (TRN).

1 Introduction

Research has demonstrated the existence of recurring small graph structures in many types of networks from domains as diverse as computer science and biology [8]. These recurring subgraph patterns are variously called *network motifs*, *graphlets* or more simply *subgraphs*. It has also been shown that complex networks can be compared and classified into distinct functional families, based on their typical motifs [7].

Moreover, biomolecular networks are hierarchical structures that consist of smaller modules of interacting components [2]. Therefore, global metrics, such as degree distribution and clustering coefficient, can not be used to completely analyze their properties [9] and, as a result, local approaches have become more prominent in the study of networks structure. The **hierarchical** and **modular** nature of biological networks [5, 13] has also been elucidated. Graph motifs aggregate into larger clusters and some of the global topological characteristics of graphs originate from the local combinations of smaller subunits.

Furthermore, close investigation into the structure of transcriptional and metabolic networks of *E. coli* and *S. cerevisiae* has suggested that this combination of motifs is not random. There appears to be a type of **preferential attachment** where homologous motifs cluster together [6]. All this implies that a network's large-scale topological organisation and its local subgraph structure mutually define and predict each other and that networks need also to be evaluated beyond the level of single subgraphs, at the level of subgraphs clusters [13].

The above observations served as inspiration for the creation of a new **network model** that integrates **global** knowledge about the presence of network motifs and their **local** combinations. More specifically, global statistical knowledge about the presence of subgraphs is combined with local knowledge about the specific way in which these motifs are interconnected. Two additional ideas were incorporated to allow better integration of this knowledge into a practical analysis method. The first is linearization of the network into an **ordered sequence of motifs**. Secondly, a **probabilistic** approach was chosen that incorporates the ideas of **growth** and **preferential attachment** together with knowledge about recurring structural elements to allow both decomposition of existing graphs and composition of similar graphs.

2 The model

The translation from graph to sequence is accomplished using a probabilistic model that describes the occurrence of, and connections between *motifs*. Because of the probabilistic nature of the model a sequence of symbols describes instances of a graph family, rather than one single graph. Using the probability distributions as a central source of information, this method can be used to both *decompose* existing graphs in to sequences, and *compose* new instances starting from such a sequence.

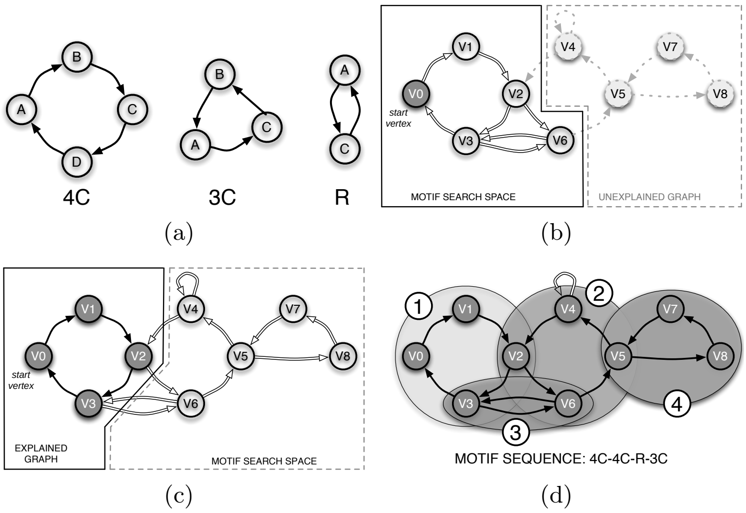
Motifs are detected in an existing graph by mapping their edges onto edges in the graph. Additional motifs are connected to already detected motifs by merging some of their vertices. As a result, each edge in the graph belongs to exactly one motif in the sequence, while graph vertices can belong to multiple motifs.

All of the information needed to construct the model is contained in a *set of motifs*. It is important to clarify that the term *motif* takes on additional meaning relative to its use in existing literature [11, 8]. Intuitively a motif can be understood to be a small graph with additional information that specifies which vertices can *attach* to other vertices and associated rules that govern the way it can connect to other motifs. These preference rules are expressed as probability distributions.

2.1 Graph decomposition

To illustrate graph decomposition, the example graph depicted in Fig. 1(d) is decomposed using the motif set in Fig. 1(a). This set becomes the alphabet of the decomposition sequence. In this example, the set consists of a 4-node cycle ($4C$), a 3-node cycle ($3C$) and a reflexive edge structure with 2 nodes and 2 edges (R). In general, the choice of motif set can be driven by domain knowledge or by graph mining techniques that compose a set based on a collection of example graphs (see also Section 3.1)

The initial step involves choosing a vertex that will serve as the starting point for motif detection. The *start vertex* can be any vertex in the source graph but



(a) (b)

(c) (d)

Fig. 1. Example graph decomposition using a sample set of motifs

its choice will influence the decomposition process and the resulting sequence. In the example, vertex $V0$ is chosen as the start vertex. Decomposition is a gradual process, during which motifs are detected in a selected area around the already explained graph. This area of the graph is referred to as the *motif search space*. The depth of the search space depends on the undirected diameter of the biggest motif in the motif set. The motif search space grows by adding edges and vertices, expanding outwards from *fringe vertices*. Initially, the only fringe vertex is the start vertex.

At each step edges and vertices within an undirected distance of E_d from the fringe vertices are added to the search space. In our example the expansion depth of the motif search space at each iteration is two, the largest diameter of any motif in the set. This means that, following the initial expansion from the start vertex $V0$, the *motif search space* contains vertices $V0$ through $V3$, $V6$ and all edges connecting them. Vertices $V2$ and $V6$ are separated from $V0$ by two edges and are therefore not expanded further until they are explained by a motif (situation in Fig. 1(b)).

Edges and vertices that are explained by a motif become part of the *explained graph*. Every consecutive motif in the sequence is required to share its attaching vertices with the part of the graph that has already been explained. The exact number of shared vertices is a function of the motif definition.

At this point, the only matching subgraph is a $4C$ motif, which becomes the first motif in the growing sequence. The only requirement for the initial motif is that it contains the starting vertex. Vertices $V0$, $V1$, $V2$ and $V3$ now become part of the explained graph. It is important to note that only the *edges* of this first motif are removed from the search space. The decomposition is edge-based:

an edge in the source graph can be explained by only one motif in the sequence, while vertices connect motifs and are shared between them.

Starting from this situation, a new expansion is done, and vertices $V4$, $V5$, $V7$ and $V8$ are added to the search space, along with their interconnecting edges (Fig. 1(c)).

Both a second $4C$ motif or an R motif can now be mapped onto the currently unexplained edges. Both would share one vertex ($V2$ and $V3$ respectively) with the first $4C$ motif. In principle, either one could become the next motif in the sequence. To make the choice between *candidate motifs*, it is necessary to introduce the mechanism that can express relative preference for each of the candidates. This is accomplished using the concept of *preferential attachment.*

Preferential attachment rules (Fig. 2) provide a way to evaluate the likelihood of candidate *motifs* and their interconnections, during both graph composition and decomposition. Three essential concepts are combined to determine which candidate should become the next motif in the sequence. Each aspect is described using a probability distribution. These distributions will become the central pieces of information for both graph *decomposition* and *composition.*

Motif-set prior. The first component is a prior preference for specific types of motifs in the motif set. In the example that we are discussing this is a uniform distribution over the motifs in the set because no particular preference has been assigned to any of them.

Motif-Vertex preference. As described in the example, motifs are regarded in the context of their connection to adjacent motifs in the graph. Such connections give rise to a partitioning of a motif's vertices into a set of *attaching vertices* $\mathcal{A}$ and a set of *non-attaching vertices. Attaching vertices* serve as connection points to already explained motifs, while *non-attaching vertices* are mapped onto previously unexplained vertices in the current *motif search space.* Every *attaching vertex* has an associated probability distribution over all possible vertices in the motif set, indicating its affinity for specific motifs and vertices belonging to them. The distribution effectively defines which motifs and vertices are preferred candidates for attachment.

Sequence Distance Rule. The final concept is the *sequence distance rule.* Each attaching vertex of a motif contains an additional probability distribution governing its affinity for a target based on a concept of distance in the (de)composition sequence. Differential preference can be given to attachment between motifs depending on the number of motifs in the sequence that separate them.

During decomposition, the likelihood that any newly discovered motif becomes the next one in the sequence is evaluated in the context of the already explained graph and the growing sequence. A newly detected candidate motif m is connected by its attaching vertices to a set of already detected motifs. Each attaching vertex $v \in \mathcal{A}$ is merged with a vertex v' belonging to a motif m' earlier in the sequence. The distance $d(m, m')$ between motif m and motif m' is defined as the number of motifs separating them in the sequence.

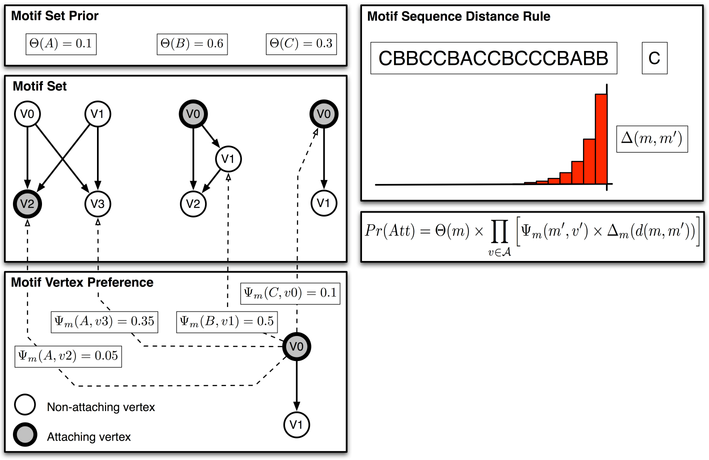


Fig. 2. Example motif set and preferential attachment rules (motif prior $\mathit{\Theta}$, motif-vertex preference $\mathit{\Psi}_m$ and sequence distance rule $\mathit{\Delta}_m$) for one of the motifs.

Definition 1. ***Probability of attachment.*** *The probability of the attachment is the product over all attaching vertices of the three preferential attachment components: the motif prior $\mathit{\Theta}$, the motif-vertex preference $\mathit{\Psi}_m$ and the sequence distance rule $\mathit{\Delta}_m$:*

$$Pr(Att) = \mathit{\Theta}(m) \times \prod_{v \in \mathcal{A}} \Big[\mathit{\Psi}_m(m', v') \times \mathit{\Delta}_m(d(m, m'))\Big]$$

Back in our example, the choice between a $4C$ and an R motif is resolved by calculating the likelihood for each attachment. First, it is necessary to check whether every *attaching vertex* is mapped onto an already explained vertex in the discovered subgraph. If we accept that the example $4C$ motif has only one attaching vertex and the R motif has two, the only valid decomposition sequence is $4C - 4C - R$. Should the *motif set* contain multiple variants of the $4C$ and R motifs with a different number of attaching vertices, both candidates could be valid. Their *preferential attachment rules* would then determine their order in the sequence. If candidates are equally likely, the choice is made randomly.

Continuing this procedure, and given that the correct sequence is $4C - 4C - R - 3C$, all vertices in the source graph have now been explained. However, this still leaves one edge unexplained, the self-loop at $V4$. This demonstrates that even with a well-chosen motif set that quite accurately captures the structural characteristics of the source graph it may not be possible to completely decompose a graph. To deal with this, *glue motifs* can be introduced into the motif set to collect edges or vertices that can not otherwise be mapped with the conventional motifs. An adequate choice of the motif set would limit the necessity for *glue motifs*.

2.2 Graph composition

Graph composition is governed by the same *preferential attachment rules* as graph decomposition. Starting from a sequence of motifs a graph instance is generated by probabilistically adding edges and vertices as determined by the motifs in the sequence. Attaching motif vertices are merged with graph vertices created by motifs earlier in the sequence. Non-attaching motif vertices create new graph vertices. The number of edges in the resulting graph equals the sum of the number of edges of all motifs in the sequence.

When adding the next motif in the sequence, every attaching vertex has to be mapped onto a graph vertex. To do so, all of the graph vertices are evaluated as potential candidates, using the *preferential attachment rules*. Two additional constraints guide this process: an attachment may not introduce parallel edges and two vertices belonging to the same motif may not be merged, since this would fundamentally alter the structure of the motif.

The likelihood for each valid attachment point is calculated and an ultimately one is chosen using roulette-wheel selection.

3 Experimental evaluation

3.1 Learning

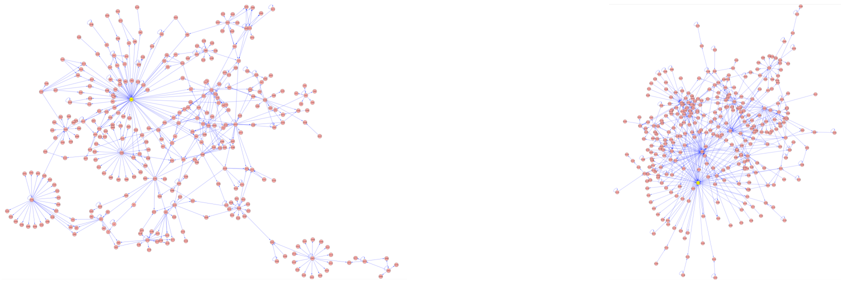
In order to use the system in a new setting — for example the characterization of gene regulatory networks — it is necessary to construct a motif set that adequately characterizes the desired graph family. Because for complex networks it is not feasible to construct the model manually, a *machine learning* was chosen in this work.

Given a training set of positive examples of a certain graph class, an evolutionary algorithm [4] was used to learn a motif set that can generate similar graphs and classify graphs as belonging to this class. In one experiment the largest connected component of the *E. coli* transcriptional regulatory network as described by [11] was used as a training set (Fig. 3(a)). Fig. 3(b) shows an example graph composed with the learned model.

3.2 Classification

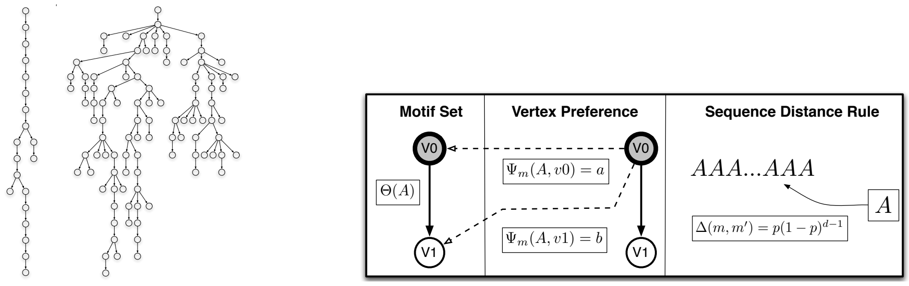
Starting with the same motif set (Fig. 4(c), $a = 0$, $b = 1$) a variety of *trees* was composed from a motif sequence by changing the sequence distance rule. When using a geometric distribution with $p = 0.9$, the trees that are generated are very chain-like, with very few and short branches (Fig. 4a). In this case it is extremely likely that new motifs in the sequence attach to adjoining motifs while with $p = 0.1$ attachments further up the chain are much more likely, resulting in the creation of many branches(Fig. 4b).

One thousand chain-like and one thousand highly-branched trees were generated from a one hundred motif sequence with appropriate parameters. Every tree was decomposed, starting from the root vertex, using the motif set that



(a) (b)

Fig. 3. (a) Largest connected component of the *E. coli* TRN used as training set for learning the TRN model. (b) Example of a graph composed with the learned model.



(a) (b) (c)

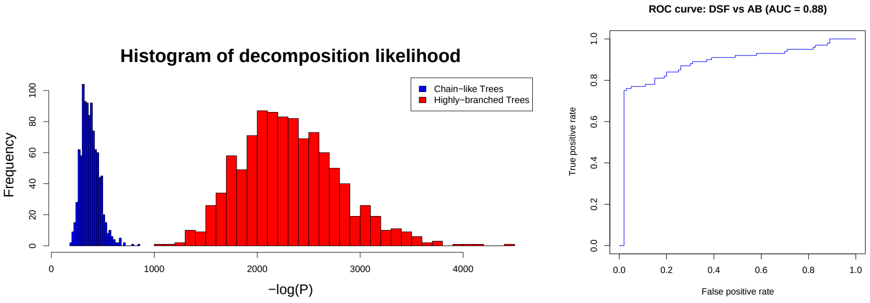
Fig. 4. A chain-like tree (a), characterized by a low number of very short branches and a highly-branched tree (b). Motif set used to generate them (c).

generates chain-like trees. As explained in Section 2.1, during the decomposition, the likelihood that any newly discovered motif becomes the next one in the sequence is evaluated in the context of the already explained graph and the growing sequence. The sum of these likelihoods, expressed as negative log-probabilities (Def. 2), can be interpreted as an overall score for the plausibility of the decomposition.

Definition 2. ***Likelihood of decomposition.*** *The likelihood L of decomposition of a graph G, given a motifset $\mathcal{M}$ into a motif sequence $S(\mathcal{M})$ is defined as*

$$L = \sum_{a \in S(\mathcal{M})} -log(Pr(Att)_a)$$

This likelihood can also be seen as a measure for the probability of generating the decomposed graph from the sequence, given the specific motif set. Fig. 5(a) shows the histogram of the resulting log-probability scores for all decompositions. As expected, the likelihood of the decomposition is much higher for the chain-like trees than for the highly-branched trees. The introduction of new branches results in lower scores because new motifs do not attach to the most recently discovered motifs, but to motifs further away in the sequence, which is unlikely given the geometric distribution with $p = 0.9$.



(a)

(b)

Fig. 5. (a) Histogram of decomposition likelihood for two families of trees: *chain-like (blue)* and *highly-branched (red)*. (b) Receiver operating characteristic (ROC) curve for two-way classification between AB and DSF graphs.

A similar experiment using a motif set learned from the *E. coli* TRN [11] was used to decompose a series of random graphs generated with the Albert-Barabási (AB)[1] and the directed scale free (DSF) [3] models. The likelihood of the decomposed sequence was then used as a score for the overall plausibility of the decomposition. Fig. 5(b) shows that it is possible to distinguish these different graph classes using the learned model.

4 Conclusions

This paper presented a model that allows characterization of graph families through a (de)composition method based on probabilistic sequences of motifs. Given a motif set, a sequence can probabilistically produce many graphs by sequentially combining the motifs. Both decomposition and composition are governed by the same probability distributions, that dictate the order of motif detection or combination.

The feasibility of using a machine learning approach to construct a suitable motif set for a new family of graphs was demonstrated. These learned motif sets can then be used to distinguish between different classes of graphs.

Acknowledgments This work was supported by an SBO grant (IWT-60045) of the Flemish Institute supporting Scientific-Technological Research in industry (IWT) and has been partially funded by the EU contract IQ FP6-516169. The authors also wish to thank Bart Goethals, Kris Laukens, Bart Naudts and Piet van Remortel for their valuable input and feedback concerning this work.

References

1. R. Albert and A. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74:4797, 2002.
2. A.-L. Barabási and Z. N. Oltvai. Network biology: understanding the cell's functional organization. *Nat Rev Genet*, 5(2):101113, 2004.
3. B. Bollobás, C. Borgs, C. Chayes, and O. Riordan. Directed scale-free graphs. In *Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithms*, page 132139, 2003.
4. D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley Professional, Reading, MA, USA, 1989.
5. H.-W. Ma, J. Buer, and A.-P. Zeng. Hierarchical structure and modules in the *Escherichia coli* transcriptional regulatory network revealed by a new top-down approach. *BMC Bioinformatics*, 5:199, 2004.
6. H.-W. Ma and A.-P. Zeng. The connectivity structure, giant strong component and centrality of metabolic networks. *Bioinformatics*, 19(11):14231430, 2003.
7. R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. Ayzenshtat, M. Sheffer, and U. Alon. Superfamilies of evolved and designed networks. *Science*, 303(5663):15381542, 2004.
8. R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: Simple building blocks of complex networks. *Science*, 298:824827, 2002.
9. N. Pržulj, D. G. Corneil, and I. Jurisica. Modeling interactome: scale-free or geometric? *Bioinformatics*, 20(18):35083515, 2004.
10. N. Pržulj. Biological network comparison using graphlet degree distribution. *Bioinformatics*, 23(2):e177–e183, 2007.
11. S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon. Network motifs in the transcriptional regulation network of *Escherichia coli. Nature Genetics*, 31:6468, 2002.
12. T. Van den Bulcke, K. Van Leemput, B. Naudts, P. van Remortel, H. Ma, A. Verschoren, B. De Moor, and K. Marchal. SynTReN: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. *BMC Bioinformatics*, 7:43, 2006.
13. A. Vazquez, R. Dobrin, D. Sergi, J. Eckmann, Z. Oltvai, and A. Barabási. The topological relationship between the large-scale attributes and local interaction patterns of complex networks. *Proceedings of the National Academy of Sciences*, 101(52):1794017945, 2004.

Sampling for Gaussian Process Inference

Michalis K. Titsias, Neil Lawrence and Magnus Rattray

School of Computer Science, University of Manchester, UK

Abstract. Sampling functions in Gaussian process (GP) models is challenging because of the highly correlated posterior distribution. We describe an efficient Markov chain Monte Carlo algorithm for sampling from the posterior process of the GP model. This algorithm uses control variables which are auxiliary function values that provide a low dimensional representation of the function. At each iteration, the algorithm proposes new values for the control variables and generates the function from the conditional GP prior. The control variable input locations are found by continuously minimizing an objective function. We use this algorithm to estimate the parameters of a differential equation model of gene regulation.

1 Introduction

Gaussian processes (GPs) are used for Bayesian non-parametric estimation of unobserved or latent functions. In regression problems with Gaussian likelihoods, inference in GP models is analytically tractable, while for classification deterministic approximate inference algorithms are widely used [5]. However, in recent applications of GP models in systems biology [1] that require the estimation of ordinary differential equation models [2, 7, 3], the development of deterministic approximations is difficult since the likelihood can be highly complex. In this paper, we consider Markov chain Monte Carlo (MCMC) algorithms for inference in GP models. An advantage of MCMC over deterministic approximate inference is that it provides an arbitrarily precise approximation to the posterior distribution in the limit of long runs. Another advantage is that the sampling scheme will often not depend on details of the likelihood function, and is therefore very generally applicable.

In order to benefit from the advantages of MCMC it is necessary to develop an efficient sampling strategy. This has proved to be particularly difficult in many GP applications, because the posterior distribution describes a highly correlated high-dimensional variable. Thus simple MCMC sampling schemes such as Gibbs sampling can be very inefficient. In this contribution we describe an efficient MCMC algorithm for sampling from the posterior process of a GP model which constructs the proposal distributions by utilizing the GP prior. This algorithm uses control variables which are auxiliary function values. At each iteration, the algorithm proposes new values for the control variables and samples the function by drawing from the conditional GP prior. The control variables are highly informative points that provide a low dimensional representation of the function. The control input locations are found by continuously minimizing an objective function. The objective function used is the expected least squares error of reconstructing the function values from the control variables, where the expectation is over the GP prior. We apply the algorithm to inference in a systems biology model where a set of genes is regulated by a transcription factor protein [3].

2 Sampling algorithms for Gaussian Process models

In a GP model we assume a set of inputs $(\mathbf{x}_1, \ldots, \mathbf{x}_N)$ and a set of function values $\mathbf{f} = (f_1, \ldots, f_N)$ evaluated at those inputs. A Gaussian process places a prior on $\mathbf{f}$ which is a N-dimensional Gaussian distribution so that $p(\mathbf{f}) = N(\mathbf{y}|\boldsymbol{\mu}, K)$. The mean $\boldsymbol{\mu}$ is typically zero and the covariance matrix K is defined by the kernel function $k(\mathbf{x}_n, \mathbf{x}_m)$ that depends on parameters $\boldsymbol{\theta}$. GPs are widely used for supervised learning [5] in which case we have a set of observed pairs $(\mathbf{y}_i, \mathbf{x}_i)$, where $i = 1, \ldots, N$, and we assume a likelihood model $p(\mathbf{y}|\mathbf{f})$ that depends on parameters $\boldsymbol{\alpha}$. For regression or classification problems, the latent function values are evaluated at the observed inputs and the likelihood factorizes according to $p(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^{N} p(y_i|f_i)$. However, for other type of applications, such as modelling latent functions in ordinary differential equations, the above factorization is not applicable. Assuming that we have obtained suitable values for the model parameters $(\boldsymbol{\theta}, \boldsymbol{\alpha})$ inference over $\mathbf{f}$ is done by applying Bayes rule:

$$p(\mathbf{f}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{f})p(\mathbf{f}). \tag{1}$$

For regression, where the likelihood is Gaussian, the above posterior is a Gaussian distribution that can be obtained using simple algebra. When the likelihood $p(\mathbf{y}|\mathbf{f})$ is non-Gaussian, computations become intractable and we need to carry out approximate inference.

The MCMC algorithm we consider is the general Metropolis-Hastings (MH) algorithm [6]. Suppose we wish to sample from the posterior in eq. (1). The MH algorithm forms a Markov chain. We initialize $\mathbf{f}^{(0)}$ and we consider a proposal distribution $Q(\mathbf{f}^{(t+1)}|\mathbf{f}^{(t)})$ that allows us to draw a new state given the current state. The new state is accepted with probability $\min(1, A)$ where

$$A = \frac{p(\mathbf{y}|\mathbf{f}^{(t+1)})p(\mathbf{f}^{(t+1)})}{p(\mathbf{y}|\mathbf{f}^{(t)})p(\mathbf{f}^{(t)})} \frac{Q(\mathbf{f}^{(t)}|\mathbf{f}^{(t+1)})}{Q(\mathbf{f}^{(t+1)}|\mathbf{f}^{(t)})}. \tag{2}$$

To apply this generic algorithm, we need to choose the proposal distribution Q. For GP models, finding a good proposal distribution is challenging since $\mathbf{f}$ is high dimensional and the posterior distribution can be highly correlated.

To motivate the algorithm presented in section 2.1, we discuss two extreme options for specifying the proposal distribution Q. One simple way to choose Q is to set it equal to the GP prior $p(\mathbf{f})$. This gives us an independent MH algorithm [6]. However, sampling from the GP prior is very inefficient as it is unlikely to obtain a sample that will fit the data. Thus the Markov chain will get stuck in the same state for thousands of iterations. On the other hand, sampling from the prior is appealing because any generated sample satisfies the smoothness requirement imposed by the covariance function. Functions drawn from the posterior GP process should satisfy the same smoothness requirement as well.

The other extreme choice for the proposal, that has been considered in [4], is to apply Gibbs sampling where we iteratively draw samples from each posterior conditional density $p(f_i|\mathbf{f}_{-i}, \mathbf{y})$ with $\mathbf{f}_{-i} = \mathbf{f} \setminus f_i$. However, Gibbs sampling can be extremely slow for densely discretized functions, as in the regression problem of Figure 1, where the posterior GP process is highly correlated. To clarify this, note that the variance of the posterior conditional $p(f_i|\mathbf{f}_{-i}, \mathbf{y})$ is smaller or equal to the variance of the conditional GP prior $p(f_i|\mathbf{f}_{-i})$. However, $p(f_i|\mathbf{f}_{-i})$ may already have a tiny variance caused by the conditioning on all remaining latent function values. For the one-dimensional example in Figure 1, Gibbs sampling is practically not applicable. We will further study this issue in section 4.

A similar algorithm to Gibbs sampling can be expressed by using the sequence of the conditional densities $p(f_i|\mathbf{f}_{-i})$ as a proposal distribution for the MH algorithm[1]. We call this algorithm the Gibbs-like algorithm. This algorithm can exhibit a high acceptance rate, but it is inefficient to sample from highly correlated functions.

2.1 Sampling using control variables

Let $\mathbf{f}_c$ be a set of M auxiliary function values that are evaluated at inputs X_c and drawn from the GP prior. We call $\mathbf{f}_c$ the control variables and their meaning is analogous to the active or inducing variables used in sparse GP models; see e.g. [5]. To compute the posterior $p(\mathbf{f}|\mathbf{y})$ based on control variables we use the expression

$$p(\mathbf{f}|\mathbf{y}) = \int_{\mathbf{f}_c} p(\mathbf{f}|\mathbf{f}_c, \mathbf{y}) p(\mathbf{f}_c|\mathbf{y}) d\mathbf{f}_c. \tag{3}$$

Assuming that $\mathbf{f}_c$ is highly informative about $\mathbf{f}$, so that $p(\mathbf{f}|\mathbf{f}_c, \mathbf{y}) \simeq p(\mathbf{f}|\mathbf{f}_c)$, we can approximately sample from $p(\mathbf{f}|\mathbf{y})$ in a two-stage manner: firstly sample the control variables from $p(\mathbf{f}_c|\mathbf{y})$ and then generate $\mathbf{f}$ from the conditional prior $p(\mathbf{f}|\mathbf{f}_c)$. This scheme can allow us to introduce a MH algorithm, where we need to specify only a proposal distribution $q(\mathbf{f}_c^{(t+1)}|\mathbf{f}_c^{(t)})$, that will mimic sampling from $p(\mathbf{f}_c|\mathbf{y})$, and always sample $\mathbf{f}$ from the conditional prior $p(\mathbf{f}|\mathbf{f}_c)$. The whole proposal distribution takes the form

$$Q(\mathbf{f}^{(t+1)}, \mathbf{f}_c^{(t+1)}|\mathbf{f}^{(t)}, \mathbf{f}_c^{(t)}) = p(\mathbf{f}^{(t+1)}|\mathbf{f}_c^{(t+1)}) q(\mathbf{f}_c^{(t+1)}|\mathbf{f}_c^{(t)}). \tag{4}$$

Each proposed sample is accepted with probability $\min(1, A)$ where A is given by

$$A = \frac{p(\mathbf{y}|\mathbf{f}^{(t+1)}) p(\mathbf{f}_c^{(t+1)})}{p(\mathbf{y}|\mathbf{f}^{(t)}) p(\mathbf{f}_c^{(t)})} . \frac{q(\mathbf{f}_c^{(t)}|\mathbf{f}_c^{(t+1)})}{q(\mathbf{f}_c^{(t+1)}|\mathbf{f}_c^{(t)})}. \tag{5}$$

[1] Thus we replace the proposal distribution $p(f_i|\mathbf{f}_{-i}, \mathbf{y})$ with the prior conditional $p(f_i|\mathbf{f}_{-i})$.

The usefulness of the above sampling scheme stems from the fact that the control variables can form a low-dimensional representation of the function. Assuming that these variables are much fewer than the points in $\mathbf{f}$, the sampling is mainly carried out in the low dimensional space. In section 2.2 we describe how to select the number M of control variables and the inputs X_c so as $\mathbf{f}_c$ becomes highly informative about $\mathbf{f}$. In the remainder of this section we discuss how we set the proposal distribution $q(\mathbf{f}_c^{(t+1)}|\mathbf{f}_c^{(t)})$.

A suitable choice for q is to use a Gaussian distribution with diagonal or full covariance matrix. The covariance matrix can be adapted during the burn-in phase of MCMC in order to increase the acceptance rate. Although this scheme is general, it has practical limitations. Firstly, tuning a full covariance matrix is time consuming and in our case this adaption process must be carried out simultaneously with searching for an appropriate set of control variables. Also, since the terms involving $p(\mathbf{f}_c)$ do not cancel out in the acceptance probability in eq. (5), using a diagonal covariance for the q distribution has the risk of proposing control variables that may not satisfy the GP prior smoothness requirement. To avoid these problems, we define q by utilizing the GP prior. According to eq. (3) a suitable choice for q must mimic the sampling from the posterior $p(\mathbf{f}_c|\mathbf{y})$. Given that the control points are far apart from each other, Gibbs sampling in the control variables space can be efficient. However, iteratively sampling f_{c_i} from the conditional posterior $p(\mathbf{f}_{c_i}|\mathbf{f}_{c_{-i}}, \mathbf{y}) \propto p(\mathbf{y}|\mathbf{f}_c)p(f_{c_i}|\mathbf{f}_{c_{-i}})$, where $\mathbf{f}_{c_{-i}} = \mathbf{f}_c \setminus f_{c_i}$ is intractable for non-Gaussian likelihoods[2]. An attractive alternative is to use a Gibbs-like algorithm where each f_{c_i} is drawn from the conditional GP prior $p(\mathbf{f}_{c_i}^{(t+1)}|\mathbf{f}_{c-i}^{(t)})$ and is accepted using the MH step. More specifically, the proposal distribution draws a new $f_{c_i}^{(t+1)}$ for a certain control variable i from $p(f_{c_i}^{(t+1)}|\mathbf{f}_{c_{-i}}^{(t)})$ and generates the function $\mathbf{f}^{(t+1)}$ from $p(\mathbf{f}^{(t+1)}|f_{c_i}^{(t+1)}, \mathbf{f}_{c_{-i}}^{(t)})$. The sample $(f_{c_i}^{(t+1)}, \mathbf{f}^{(t+1)})$ is accepted using the MH step. This scheme of sampling the control variables one-at-a-time and resampling $\mathbf{f}$ is iterated between different control variables. A complete iteration of the algorithm consists of a full scan over all control variables. The acceptance probability A in eq. (5) becomes the likelihood ratio and the prior smoothness requirement is always satisfied.

Although the control variables are sampled one-at-at-time, $\mathbf{f}$ can still be drawn with a considerable variance. To clarify this, note that when the control variable f_{c_i} changes the effective proposal distribution for $\mathbf{f}$ is $p(\mathbf{f}^{t+1}|\mathbf{f}_{c_{-i}}^{(t)}) = \int_{f_{c_i}^{(t+1)}} p(\mathbf{f}^{t+1}|f_{c_i}^{(t+1)}, \mathbf{f}_{c_{-i}}^{(t)})p(f_{c_i}^{(t+1)}|\mathbf{f}_{c_{-i}}^{(t)})df_{c_i}^{(t+1)}$, which is the conditional GP prior given all the control points apart from the current point f_{c_i}. This conditional prior can have considerable variance close to f_{c_i} and in all regions that are not close to the remaining control variables. The iteration over different control variables allow $\mathbf{f}$ to be drawn with a considerable variance everywhere in the input space.

[2] This is because we need to integrate out $\mathbf{f}$ in order to compute $p(\mathbf{y}|\mathbf{f}_c)$.

2.2 Selection of the control variables

To apply the previous algorithm we need to select the number, M, of the control points and the associated inputs X_c. X_c must be chosen so that knowledge of $\mathbf{f}_c$ can determine $\mathbf{f}$ with small error. The prediction of $\mathbf{f}$ given $\mathbf{f}_c$ is equal to $K_{f,c}K_{c,c}^{-1}\mathbf{f}_c$ which is the mean of the conditional prior $p(\mathbf{f}|\mathbf{f}_c)$. A suitable way to search over X_c is to minimize the reconstruction error $||\mathbf{f} - K_{f,c}K_{c,c}^{-1}\mathbf{f}_c||^2$ averaged over any possible value of $(\mathbf{f}, \mathbf{f}_c)$:

$$G(X_c) = \int_{\mathbf{f},\mathbf{f}_c} ||\mathbf{f} - K_{f,c}K_{c,c}^{-1}\mathbf{f}_c||^2 p(\mathbf{f}|\mathbf{f}_c)p(\mathbf{f}_c)d\mathbf{f}d\mathbf{f}_c = \mathrm{Tr}(K_{f,f} - K_{f,c}K_{c,c}^{-1}K_{f,c}^T).$$

The quantity inside the trace is the covariance matrix of $p(\mathbf{f}|\mathbf{f}_c)$ and thus $G(X_c)$ is the total variance of this distribution. We can minimize $G(X_c)$ w.r.t. X_c using continuous optimization. Note that $G(X_c)$ is nonnegative and when it becomes zero, $p(\mathbf{f}|\mathbf{f}_c)$ becomes a delta function.

To find the number M of control points we minimize $G(X_c)$ by incrementally adding control variables until the total variance of $p(\mathbf{f}|\mathbf{f}_c)$ becomes smaller than a certain percentage of the total variance of the prior $p(\mathbf{f})$. 5% was the threshold used in all our experiments. Then we start the simulation and we observe the acceptance rate of the Markov chain. According to standard heuristics [6] which suggest that desirable acceptance rates of MH algorithms are around 1/4, we require a full iteration of the algorithm (a complete scan over the control variables) to have an acceptance rate larger than 1/4. When for the current set of control inputs X_c the chain has a low acceptance rate, it means that the variance of $p(\mathbf{f}|\mathbf{f}_c)$ is still too high and we need to add more control points in order to further reduce $G(X_c)$. The process of observing the acceptance rate and adding control variables is continued until we reach the desirable acceptance rate.

3 Transcriptional regulation

We consider a small biological sub-system where a set of target genes are regulated by one transcription factor (TF) protein. Ordinary differential equations (ODEs) can provide an useful framework for modelling the dynamics in these biological networks [1, 2, 7, 3]. The concentration of the TF and the gene specific kinetic parameters are typically unknown and need to be estimated by making use of a set of observed gene expression levels. We use a GP prior to model the unobserved TF activity, as proposed in [3], and apply full Bayesian inference based on the MCMC algorithm presented previously.

Barenco et al. [2] introduce a linear ODE model for gene activation from TF. This approach was extended in [7, 3] to account for non-linear models. The general form of the ODE model for transcription regulation with a single TF has the form

$$\frac{dy_j(t)}{dt} = B_j + S_j g(f(t)) - D_j y_j(t), \tag{6}$$

where the changing level of a gene j's expression, $y_j(t)$, is given by a combination of basal transcription rate, B_j, sensitivity, S_j, to its governing TF's activity, $f(t)$, and the decay rate of the mRNA, D_j. The differential equation can be solved for $y_j(t)$ giving

$$y_j(t) = \frac{B_j}{D_j} + A_j e^{-D_j t} + S_j e^{-D_j t} \int_0^t g(f(u)) e^{D_j u} du, \tag{7}$$

where A_j term arises from the initial condition. Due to the non-linearity of the g function that transforms the TF, the integral in the above expression is not analytically obtained. However, numerical integration can be used to accurately approximate the integral with a dense grid $(u_i)_{i=1}^P$ of points in the time axis and evaluating the function at the grid points $f_p = f(u_p)$. In this case the integral in the above equation can be written $\sum_{p=1}^{P_t} w_p g(f_p) e^{D_j u_p}$ where the weights w_p arise from the numerical integration method used and, for example, can be given by the composite Simpson rule.

The TF concentration $f(t)$ in the above system of ODEs is a latent function that needs to be estimated. Additionally, the kinetic parameters of each gene $\boldsymbol{\alpha}_j = (B_j, D_j, S_j, A_j)$ are unknown and also need to be estimated. To infer these quantities we use mRNA measurements (obtained from microarray experiments) of N target genes at T different time steps. Let y_{jt} denote the observed gene expression level of gene j at time t and let $\mathbf{y} = \{y_{jt}\}$ collect together all these observations. Assuming a Gaussian noise for the observed gene expressions the likelihood of our data has the form

$$p(\mathbf{y}|\mathbf{f}, \{\boldsymbol{\alpha}_j\}_{j=1}^N) = \prod_{j=1}^N \prod_{t=1}^T p(y_{jt}|\mathbf{f}_{1\leq p \leq P_t}, \boldsymbol{\alpha}_j), \tag{8}$$

where each probability density in the above product is a Gaussian with mean given by eq. (7) and $\mathbf{f}_{1\leq p\leq P_t}$ denotes the TF values up to time t. Notice that this likelihood is non-Gaussian due to the non-linearity of g. Further, this likelihood does not have a factorized form, as in the regression and classification cases, since an observed gene expression depends on the protein concentration activity in all previous times points. Also note that the discretization of the TF in P time points corresponds to a very dense grid, while the gene expression measurements are sparse, i.e. $P \gg T$.

To apply full Bayesian inference in the above model, we need to define prior distributions over all unknown quantities. The protein concentration $\mathbf{f}$ is a positive quantity, thus a suitable prior is to consider a GP prior for $\log \mathbf{f}$. The kinetic parameters of each gene are all positive scalars. Those parameters are given vague gamma priors. Sampling the GP function is done exactly as described in section 2; we have only to plug in the likelihood from eq. (8) in the MH step. Sampling from the kinetic parameters is carried using Gaussian proposal distributions with diagonal covariance matrices that sample the positive kinetic parameters in the log space.

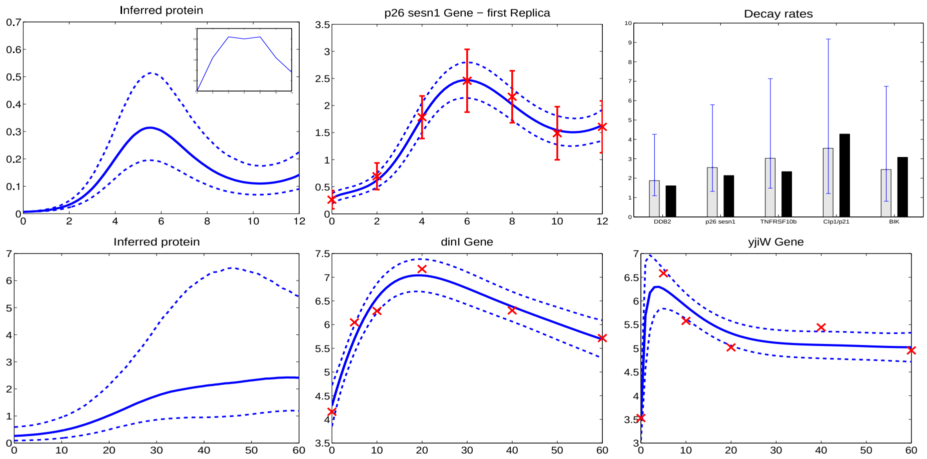


Fig. 1: **First row:** The left plot shows the inferred TF concentration for p53; the small plot on top-right shows the ground-truth protein concentration obtained by a *Western blot* experiment [2]. The middle plot shows the predicted expression of a gene obtained by the estimated ODE model; red crosses correspond to the actual gene expression measurements. The right-hand plot shows the estimated decay rates for all 5 target genes used to train the model. Grey bars display the parameters found by MCMC and black bars the parameters found in [2] using a linear ODE model. **Second row**: The left plot shows the inferred TF concentration for LexA. Predicted expression profiles of two target genes are shown in the remaining two plots. Error bars in all plots correspond to 95% credibility intervals.

4 Experiments

We consider two experiments where we apply the algorithm using control variables to infer the protein concentration of TFs that activate or repress a set of target genes. The latent function in these problems is always one-dimensional and densely sampled. Gibbs sampling schemes in such cases are extremely inefficient while the algorithm using control variables efficiently samples from the GP posterior process.

We first consider the TF p53 which is a tumour repressor activated during DNA damage. Seven samples of the expression levels of five target genes in three replicas are collected as the raw time course data. The non-linear activation of the protein follows the Michaelis Menten kinetics inspired response [1] that allows saturation effects to be taken into account so as $g(f(t)) = \frac{f(t)}{\gamma_j + f(t)}$ in eq. (6) where the Michaelis constant for the jth gene is given by γ_j. Note that since $f(t)$ is positive the GP prior is placed on the $\log f(t)$. To apply MCMC we discretize $\mathbf{f}$ using a grid of $P = 121$ points. During sampling, 7 control variables were needed to obtain the desirable acceptance rate. Running time was 4 hours for 5×10^5

sampling iterations plus 5×10^4 burn-in iterations. The first row of Figure 1 summarizes the estimated quantities obtained from MCMC simulation.

Next we consider the TF LexA in E.Coli that acts as a repressor. In the repression case there is an analogous Michaelis Menten model [1] where the nonlinear function g takes the form: $g(f(t)) = \frac{1}{\gamma_j + f(t)}$. Again the GP prior is placed on the log of the TF activity. We applied our method to the same microarray data considered in [7] where mRNA measurements of 14 target genes are collected over six time points. For this dataset, the expression of the 14 genes were available for $T = 6$ times. The GP function $\mathbf{f}$ was discretized using 121 points. The result for the inferred TF profile along with predictions of two target genes are shown in the second row of Figure 1. Our inferred TF profile and reconstructed target gene profiles are similar to those obtained in [7]. However, for certain genes, our model provides a better fit to the gene profile.

5 Discussion

Gaussian processes allow for inference over latent functions using a Bayesian estimation framework. In this paper, we presented an MCMC algorithm that uses control variables. We currently extend the MCMC framewowork to deal with much larger systems of ODEs with multiple interacting transcription factors.

References

1. U. Alon. *An Introduction to Systems Biology: Design Principles of Biological Circuits.* Chapman and Hall/CRC, 2006.
2. M. Barenco, D. Tomescu, D. Brewer, J. Callard, R. Stark, and M. Hubank. Ranked prediction of p53 targets using hidden variable dynamic modeling. *Genome Biology*, 7(3), 2006.
3. N. D. Lawrence, G. Sanguinetti, and M. Rattray. Modelling transcriptional regulation using Gaussian processes. In B. Scholkopf, J. C. Platt, and T. Hofmann, editors, *Advances in Neural Information Processing Systems, 19.* MIT Press, 2007.
4. R. M. Neal. Monte Carlo implementation of Gaussian process models for Bayesian regression and classification. Technical report, Dept. of Statistics, University of Toronto, 1997.
5. C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning.* MIT Press, 2006.
6. C. P. Robert and G. Casella. *Monte Carlo Statistical Methods.* Springer-Verlag, 2nd edition, 2004.
7. S. Rogers, R. Khanin, and M. Girolami. Bayesian model-based inference of transcription factor activity. *BMC Bioinformatics*, 8(2), 2006.

Using mRNA Secondary Structure Predictions Improves Recognition of Known Yeast Functional uORFs

Selpi[1]*, Christopher H. Bryant[2], and Graham J.L. Kemp[3]

[1] School of Computing, The Robert Gordon University,
St. Andrew Street, Aberdeen, AB25 1HG, UK
[2] School of Computing, Science and Engineering, Newton Building,
University of Salford, Salford, Greater Manchester, M5 4WT, UK
[3] Department of Computer Science and Engineering,
Chalmers University of Technology, SE-412 96 Göteborg, Sweden

Abstract. We are interested in using inductive logic programming (ILP) to generate rules for recognising functional upstream open reading frames (uORFs) in the yeast *Saccharomyces cerevisiae*. This paper empirically investigates whether providing an ILP system with predicted mRNA secondary structure can increase the performance of the resulting rules. Two sets of experiments, with and without mRNA secondary structure predictions as part of the background knowledge, were run. For each set, stratified 10-fold cross-validation experiments were run 100 times, each time randomly permuting the order of the positive training examples, and the performance of the resulting hypotheses were measured. Our results demonstrate that the performance of an ILP system in recognising known functional uORFs in the yeast *S. cerevisiae* significantly increases when mRNA secondary structure predictions are added to the background knowledge and suggest that mRNA secondary structure can affect the ability of uORFs to regulate gene expression.

1 Introduction

Uncovering the mechanisms that regulate gene expression at a system-level is an important task in systems biology. Understanding the roles of post-transcriptional regulatory elements in gene expression is one aspect of this. Upstream open reading frames (uORFs) are among the regulatory elements that can be present in the 5′ untranslated region (UTR) of messenger RNA (mRNA). In the yeast *Saccharomyces cerevisiae*, some uORFs have been well studied and it has been verified that some of these regulate gene expression (i.e. they are functional) [1–5], while a few others do not (i.e. they are non-functional) [6, 7]. The mechanism by which uORFs regulate genes is still only partially understood. This is mainly because wet-lab experiments to test whether a gene contains functional uORFs are costly and time-consuming.

[*] To whom correspondence should be addressed.

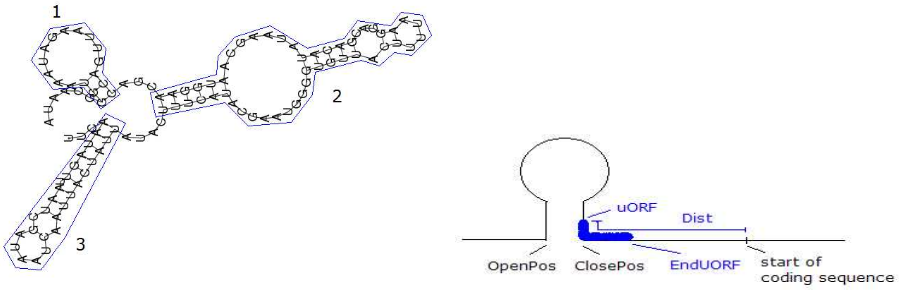


Fig. 1. Left: A predicted secondary structure of the 5′ UTR sequence and ten nucleotides of gene *YAP2* (YDR423C), made by RNAfold. The boxes have been added to show how we view the structure as three stem-loop structures. Right: Illustration of a uORF intersects with an mRNA secondary structure on the uORF's left (upstream) part.

It has been shown that inductive logic programming (ILP) can automatically generate a set of hypotheses which makes searching for novel functional uORFs (i.e. uORFs which can regulate gene expression) in the yeast *S. cerevisiae* more efficient than random sampling [8]. Those hypotheses were simple and easy to understand, but appeared to be too general. This is due not only to the limited number of positive examples and the high degree of noise in the data, two problems which cannot be easily rectified, but also due to the limited background knowledge.

In this paper, we investigate whether incorporating predicted mRNA secondary structure as background knowledge can increase the performance of the resulting hypotheses in recognising functional uORFs in the yeast *S. cerevisiae*. The type of mRNA secondary structure we consider is the stem-loop. A stem-loop is a simple RNA secondary structure motif that can occur when the transcribed sequence contains an inverted repeat sequence (see Fig. 1). Based on their study on a maize gene, Wang and Wessler [9] concluded that uORF and mRNA secondary structure regulate gene expression independently. However, the results from [10], based on studies on human genes, are rather different; the presence of secondary structure seems to affect uORFs' ability to regulate gene expression. The difference between the conclusions of [9] and those of [10] leave open the question whether mRNA secondary structure influences uORFs' ability to regulate gene expression. This motivates our study; to test whether mRNA secondary structure predictions could help in recognising known functional uORFs in the yeast *S. cerevisiae*.

The rest of this paper is organised as follows. Section 2 describes the dataset and the learning system used in this work. The experimental method, including how we incorporate mRNA secondary structure predictions as background knowledge, is detailed in Section 3. Our results are presented in Section 4. Finally, in Section 5, we discuss our main results and suggest directions for future work.

2 The Dataset and the Learning System

The same dataset that was used for training and testing in [8] was used here for training and testing. For the task of learning which uORFs regulate gene expression, positive examples are verified functional uORFs, and negative examples are verified non-functional uORFs. Since confirmed negative examples are scarce (there are only two compared to 20 positive examples) and given that there are 380 random examples (unlabelled uORFs, most of which are probably negatives), we use the positive-only setting [11] of CProgol [12] version 4.4 [13]; the same was used in [8]. CProgol is an established ILP system which uses a covering approach for hypotheses construction. CProgol has been successfully applied to many different problems, including some in bioinformatics. The positive-only setting of CProgol4.4 learns from both positive and random examples; the random examples can either be provided by the user or generated automatically by CProgol. The random examples used for our experiments here are the 380 unlabelled uORFs. We did not use the system-generated random examples because these could be less informative than unlabelled uORFs and might not represent true examples (i.e., true uORFs).

3 Methods

To enable us to test whether incorporating mRNA secondary structure predictions as background knowledge increases ILP performance when learning which uORFs in yeast are functional, we run two sets of experiments, with and without mRNA secondary structure predictions as part of the background knowledge. For each set, stratified 10-fold cross-validation experiments were run 100 times, each time with a random permutation of the order in which positive training examples are presented to the ILP system; this was done because CProgol4.4 may generate different hypotheses when given different orderings of positive training examples. The same 100 random orderings were used for both sets of experiments. Stratified 10-fold cross-validation means that the set of positive examples is divided into ten roughly equal partitions and the same is done to the set of random examples; each of these positive and random partitions are in turn used as a test set while the rest of the partitions are used as training set. Table 1 summarises our experimental procedure.

The ILP learner was instructed to learn a predicate `has_functional_role/1` from a set of training examples. Positive examples were represented as instances of the predicate `has_functional_role(X)`, where `X` is a uORF ID. A uORF ID is a composite of the systematic name of the gene to which the uORF belongs (for example, YDR423C is the systematic name of gene *YAP2*) and a uORF identifier (e.g., uORF1, uORF2, *etc.*). The definition of the hypotheses space for

Table 1. The Experimental Procedure

```
For i=1 to 100
   Randomly permute the order of examples
   Divide dataset into stratified 10 folds
      Divide set of positives into 10 equal partitions
      Divide set of randoms into 10 roughly equal partitions
      For j=1 to 10
         Concatenate partition j of positives and partition j of randoms
         to create fold j
   For each set of background knowledge
      For j=1 to 10
         Use fold j as test set
         Construct hypotheses using the other nine folds
         Use the resulting hypotheses to classify the test set
      Get the performance of stratified 10-fold cross-validation
      experiments
```

Table 2. Representation of a predicted structure shown in Figure 1. `has_stemloop(X,Y)` represents the relationship between UTR `X` and stem-loop `Y`. `stemloop(W,X,Y,Z)` states that stem-loop `W` has its opening and closing positions in `X` and `Y` bases to the coding sequence; and there are, in total, `Z` base pairs within `W`.

```
has_stemloop(YDR423C, YDR423C_sl3).    stemloop(YDR423C_sl3, 98, 71, 10).
has_stemloop(YDR423C, YDR423C_sl2).    stemloop(YDR423C_sl2, 66, 17, 13).
has_stemloop(YDR423C, YDR423C_sl1).    stemloop(YDR423C_sl1, 13, -3,  3).
```

the experiments without mRNA secondary structure predictions were the same as in Table 5 of [8].

RNAfold [14][4] was used, with its default settings, to generate mRNA secondary structure predictions from sequence data. For each of the 17 well-studied genes, the 5′ UTR sequence and the first ten nucleotides of the coding sequence was used as an input for RNAfold. The length of 5′ UTRs were taken from the European Molecular Biology Laboratory (EMBL) Nucleotide Sequence Database[5], where available, or, failing that, [1]. The output from RNAfold was transformed into Prolog predicates representing predicted mRNA secondary structure as extensional background knowledge. In this work, we view the predicted mRNA secondary structure from the highest level. This means that we do not consider a nested stem-loop as an independent stem-loop. For example, we only consider *YAP2* to have the three stem-loop structures shown in the left part of Fig. 1 and Table 2.

[4] ViennaRNA-1.6.1 was downloaded from `http://www.tbi.univie.ac.at/~ivo/RNA/`

[5] `ftp://ftp.ebi.ac.uk/pub/databases/UTR/data/5UTR.Fun_nr.dat.gz` version 16 June 2006

Table 3. Additional mode declarations used in experiments with mRNA secondary structure predictions[a].

```
:- modeb(1,is_inside_stemloop(+uORF))?
:- modeb(1,intersectleft_with_stemloop(+uORF))?
:- modeb(1,intersectright_with_stemloop(+uORF))?
:- modeb(*,has_stemloop(+uORF,-stemloop))?
:- modeb(1,stemloop(+stemloop,-pospair1,-pospair2,-numberofpairs))?
:- modeb(1,+numberofpairs=< #int)?
:- modeb(1,+numberofpairs>= #int)?
:- modeb(1,+numberofpairs= #int)?
:- modeb(1,+pospair1=< #int)?          :- modeb(1,+pospair2=< #int)?
:- modeb(1,+pospair1>= #int)?          :- modeb(1,+pospair2>= #int)?
:- modeb(1,+pospair1= #int)?           :- modeb(1,+pospair2= #int)?
```

[a] `modeb` describes the predicates to be used in a hypothesis and has the format: `modeb(RecallNumber,Template)`. `RecallNumber` specifies how many times the `Template` can be called successfully; `*` means the `Template` can be called successfully up to 100 times. `Template` is n-ary predicates, with $n \geq 1$ and each of the arguments is a *variable type* preceded by either a '+' (indicates that the argument should be an input), '-' (indicates that the argument should be an output), or '#' (indicates that the argument should be a constant). The types `uORF` and `stemloop` were declared by defining a set of instances of the predicates `uORF(X)` and `stemloop(Y)` respectively, where `X` is a uORF ID and `Y` is a stem-loop ID. The types of `pospair1`, `pospair2`, and `numberofpairs` were all defined as integer. `pospair1` and `pospair2` represent the opening and closing positions of the stemloop. `numberofpairs` represents the length of stem.

[15] and [16] suggested that the stability of a secondary structure and its distance from the coding sequence influence its ability to inhibit the translation of the coding sequence. Therefore, the predicate `stemloop/4` (see Table 3) was designed to capture both the distance (the opening and the closing positions in the right part of Fig. 1) of a predicted stem-loop structure to the coding sequence and the stability. Here, the stability was represented by the number of base pairs (the length of the stem); the longer the stem the more stable the secondary structure and the more energy is needed to unwind it. We do not use the predicted minimum free energy because of the way we view the predicted mRNA secondary structure. For example, we consider three stem-loop structures while there was only one predicted minimum free energy for the overall predicted structure shown in the left part of Fig. 1.

With the biological knowledge gained from literature, we defined several declarative rules that identify if a uORF intersects with any predicted secondary structure on the uORF's left (upstream) part (see an illustration in Fig. 1), on the uORF's right (downstream) part, or is inside any predicted secondary structure. To instruct CProgol to include mRNA secondary structure predictions in its hypothesis space, we defined additional mode declarations (Table 3). Some adjustments were made to the parameter settings used in [8] to allow CProgol to consider a larger hypotheses space. The parameter **c** (the maximum number of atoms in the body of the rules constructed) was increased from 6 to 10; **nodes** (the maximum number of nodes explored during clause searching) was increased from 7,000 to 50,000; and **h** (the maximum depth of resolutions allowed when proving) was increased from 30 (default value) to 100.

4 Results

To statistically evaluate the impact of incorporating mRNA secondary structure predictions as part of the background knowledge on the task of recognising yeast functional uORFs, we compared the relative advantage (RA) values [17, Appendix A] from 100 experiments with and without mRNA secondary structure predictions. RA was used as a performance measure in [8]. The characteristics of the data used here matched with the characteristics for which RA is claimed to be useful. The idea of using RA is to predict the cost reduction in finding functional uORFs using a recognition model compared to using random sampling. In this application domain, RA is defined as

$$RA = \frac{A}{B}$$

where

- A is the expected cost of finding one functional uORF by repeated independent random sampling from the set of possible uORFs and performing a lab analysis of each uORF;
- B is the expected cost of finding one functional uORF by repeated independent random sampling from the set of possible uORFs and analysing only those uORFs which are predicted by the learned model as functional uORFs.

In 87 experiments out of 100, the mean RA values from the experiments with mRNA secondary structure predictions are better than the mean RA values from the corresponding experiments without mRNA secondary structure predictions (see Fig. 2). The result from a *Wilcoxon Signed Rank test* shows that there was a statistically significant increase from the mean RA values from the experiments without mRNA secondary structure predictions to those from the corresponding experiments with mRNA secondary structure predictions (mean RA values: mean without=34.05, mean with=61.53, $p < 0.0005$).

The analysis made so far is based on the mean RA values from our experiments. However, RA is less well known than other performance measures such as precision, recall (also known as sensitivity), specificity, and F_1 score. Therefore, to support our analysis, we also measured the precision, recall, specificity[6], and F_1 score. We found that there were statistically significant increases in the values of precision, recall, specificity, and F_1 score from the experiments

[6] In this case, specificity measures the fraction of randoms which are predicted as randoms.

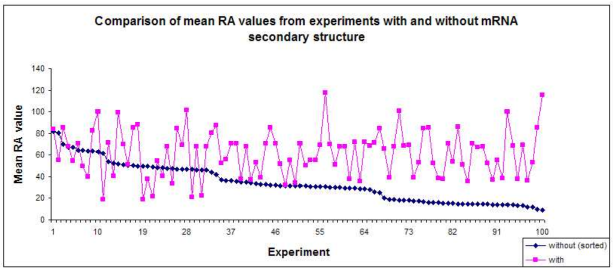


Fig. 2. Comparison of mean RA values from 100 experiments with and without mRNA secondary structure predictions. Experiments are sorted with respect to mean RA values from the experiments without mRNA secondary structure predictions. In 87 experiments, the mean RA values from experiments with mRNA secondary structure predictions are better than those from experiments without mRNA secondary structure predictions.

Table 4. Spearman's rank correlation between mean RA and other performance measures from 100 experiments with and without mRNA secondary structure predictions.

Experiment		Precision	Recall	Specificity	F_1 score
with	Mean RA	0.94	-0.02	0.73	0.74
without	Mean RA	0.91	-0.05	0.72	0.70

Note: There is no significant correlation between mean RA and recall. All other correlations are significant with $p < 0.0005$.

without mRNA secondary structure predictions to those from the corresponding experiments with mRNA secondary structure predictions (precision: mean without=0.45, mean with=0.63; recall: mean without=0.77, mean with=0.87; specificity: mean without=0.94, mean with=0.96; F_1 score: mean without=0.54, mean with=0.70; all were based on Wilcoxon Signed Ranks test with $p < 0.0005$).

Spearman's rank correlation was used to find out whether there are relationships between RA and the other measures (Table 4). We conclude that mean RA has a strong positive correlation with precision and specificity. Spearman's correlation also shows that there was a strong positive correlation between mean RA and F_1 score. This is due to the strong positive correlation between mean RA and precision, since there was no significant correlation between mean RA and recall; precision and recall are the two components used for calculating F_1 score.

The content of the hypotheses were also analysed. The hypotheses from the 10 experiments that give the 10 highest average cross-validation performances (mean RA) suggest that mRNA secondary structure influences uORFs' ability to regulate gene expression in the yeast *S. cerevisiae.* The rules also suggest that a functional uORF is likely to lie inside a stem-loop structure, or to intersect with a stem-loop structure on the uORF's left part. In our data, 17 of the 20 functional uORFs (positive examples) lie inside stem-loop structures predicted in the associated UTRs. For 3 of the 20 uORFs, their left part intersect with stem-loop structures predicted in the associated UTRs; 2 of these 3 uORFs do not lie inside stem-loop structures predicted in the associated UTRs.

5 Discussion and Future Work

Our empirical results show that the performance of an ILP system, CProgol 4.4, in recognising known functional uORFs in the yeast *S. cerevisiae* significantly increases when mRNA secondary structure predictions are added to the background knowledge (mean RA values: mean without=34.05, mean with=61.53, $p < 0.0005$). This conclusion still holds when performance is measured using precision, recall, specificity, and F_1 score, which are very well known in both machine learning and bioinformatics domains.

In this work, the background knowledge regarding mRNA secondary structure was derived from predictions made by RNAfold on the given *S. cerevisiae* sequences. However, the reliability of predictions made by RNAfold, and other similar software based on thermodynamic energy minimisation, is often questioned because each prediction is made based on a single sequence. Therefore, for future work, one could consider deriving the background knowledge from mRNA secondary structures that are predicted to be conserved among yeast species.

Here, we view the predicted mRNA secondary structure from the highest level, and do not consider a nested stem-loop as an independent stem-loop. Thus, we limited the type of background knowledge that was derived from the predicted mRNA secondary structure. It would be interesting to investigate the effect of including more detailed background knowledge of the mRNA secondary structure predictions on the ILP system's performance in recognising functional uORFs.

Acknowledgments. We thank Alex Wilson and Dr John McCall for useful discussions on statistics and experimental design.

References

1. Vilela, C., McCarthy, J.E.G.: Regulation of fungal gene expression via short open reading frames in the mRNA 5′ untranslated region. Molecular Microbiology **49**(4) (2003) 859–867
2. Vilela, C., Ramirez, C.V., Linz, B., Rodrigues-Pousada, C., McCarthy, J.E.G.: Post-termination ribosome interactions with the 5′ UTR modulate yeast mRNA stability. The EMBO Journal **18**(11) (1999) 3139–3152
3. Hinnebusch, A.G.: Translational Regulation of Yeast *GCN4*. A Window on Factors that Control Initiator-tRNA Binding to the Ribosome. J. Biol. Chem. **272**(35) (1997) 21661–21664
4. Fiaschi, T., Marzocchini, R., Raugei, G., Veggi, D., Chiarugi, P., Ramponi, G.: The 5′-untranslated region of the human muscle acylphosphatase mRNA has an inhibitory effect on protein expression. FEBS Letters **417**(1) (1997) 130–134

5. Iacono, M., Mignone, F., Pesole, G.: uAUG and uORFs in human and rodent $5'$untranslated mRNAs. Gene **349** (2005) 97–105
6. Morris, D.R., Geballe, A.P.: Upstream Open Reading Frames as Regulators of mRNA Translation. Molecular and Cellular Biology **20**(23) (2000) 8635–8642
7. Krummeck, G., Gottenöf, T., Rödel, G.: AUG codons in the RNA leader sequences of the yeast *PET* genes *CBS1* and *SCO1* have no influence on translation efficiency. Current Genetics **20** (1991) 465–469
8. Selpi, Bryant, C.H., Kemp, G.J.L., Cvijovic, M.: A First Step towards Learning which uORFs Regulate Gene Expression. Journal of Integrative Bioinformatics **3**(2) (2006) 31
9. Wang, L., Wessler, S.R.: Role of mRNA Secondary Structure in Translational Repression of the Maize Transcriptional Activator *Lc*. Plant Physiology **125**(3) (2001) 1380–1387
10. Kwon, H.S., Lee, D.K., Lee, J.J., Edenberg, H.J., ho Ahn, Y., Hur, M.W.: Post-transcriptional Regulation of Human *ADH5/FDH* and *Myf6* Gene Expression by Upstream AUG Codons. Archives of Biochemistry and Biophysics **386**(2) (2001) 163–171
11. Muggleton, S.: Learning from Positive Data. In Muggleton, S., ed.: Inductive Logic Programming Workshop. Volume 1314 of Lecture Notes in Computer Science., Springer (1996) 358–376
12. Muggleton, S.: Inverse Entailment and Progol. New Generation Computing **13**(3&4) (1995) 245–286
13. Muggleton, S., Firth, J.: CProgol4.4: a tutorial introduction. In Džeroski, S., Lavrač, N., eds.: Relational Data Mining. Springer-Verlag (2001) 160–188
14. Hofacker, I.L., Fontana, W., Stadler, P.F., Bonhoeffer, L.S., Tacker, M., Schuster, P.: Fast Folding and Comparison of RNA Secondary Structures (The Vienna RNA Package). Monatsh. Chem. **125**(2) (1994) 167–188
15. Baim, S.B., Sherman, F.: mRNA Structures Influencing Translation in the Yeast *Saccharomyces cerevisiae*. Molecular and Cellular Biology **8**(4) (1988) 1591–1601
16. Laso, M.R.V., Zhu, D., Sagliocco, F., Brown, A.J.P., Tuite, M.F., McCarthy, J.E.G.: Inhibition of Translation Initiation in the Yeast *Saccharomyces Cerevisiae* as a Function of the Stability and Position of Hairpin Structures in the mRNA Leader. The Journal of Biological Chemistry **268**(9) (1993) 6453–6462
17. Muggleton, S.H., Bryant, C.H., Srinivasan, A., Whittaker, A., Topp, S., Rawlings, C.: Are Grammatical Representations Useful for Learning from Biological Sequence Data?-A Case Study. Journal of Computational Biology **8**(5) (2001) 493–521

A Bi-ordering Approach to Linking Gene Expressions with Clinical Annotations in Cancer (Extended Abstract)

Fan Shi[1,2], Geoff MacIntyre[1,2], Christopher Leckie[1,2], Izhak Haviv[5],
Alex Boussioutas[4], and Adam Kowalczyk[1,3]
[1] National ICT Australia; [2]Department of Computer Science and Software Engineering, and
[3]Department of Electrical and Electronic Engineering,
The University of Melbourne, Parkville, Victoria 3010, Australia;
[4] Peter MacCallum Cancer Centre, St. Andrew's Place, East Melbourne, Vic. 3002, Australia;
[5] Baker IDI Heart and Diabetes Institute, 250 Kooyong Road Caulield, Vic. 3162, Australia

{shif,gmaci,caleckie}@csse.unimelb.edu.au;alex.boussioutas@petermac.org;
izhak.haviv@bakeridi.edu.au; akowalczyk@nicta.com.au

Abstract. In this paper we introduce a robust method for exploratory analysis of microarray data, which produces a number of different bi-orderings of the data, each uniquely determined by a bicluster, i.e., a pair of subsets of genes and samples. We first compare the performance of our core bi-ordering algorithm with several closely related existing biclustering algorithms on a gastric cancer dataset. We then show that the sample bi-orderings generated by our method are highly statistically significant with respect to both the sample histological annotations and biological annotations (from the Gene Ontology). We show that some of the gene modules associated with our most robust bi-orderings are closely linked to gene modules that are important for gastric cancer tumorgenesis reported in literature, while others are novel discoveries.

Keywords: biclustering, gene expression, gene ontology, gastric cancer

1 Introduction

A typical aim of exploratory analysis of genomics data is to identify potentially interesting genes or pathways that warrant further investigation. There is a critical need to streamline this type of analysis, in order to support continuing advances in high throughput genomics methods such as gene expression microarrays, which measure thousands of genes in a single assay and are the focus of this paper. Such assays provide noisy and incomplete measurements, which require sophisticated bioinformatics techniques to identify statistically and biologically significant associations between genes and relevant phenotypes of interest.

Unsupervised analysis techniques cluster data without using prior information on the labels of samples. This enables the discovery of novel histological subtypes. However, there are a number of open challenges, such as how to evaluate the statistical significance of the results. The major limitation of traditional clustering algorithms in this context is that they cluster genes into non-overlapping groups, based on the similarity of their expression across all samples. This limits the ability to find groups of genes that are "co-correlated" across only a subset of samples or participate in multiple cellular pathways. In spite of such limitations, there are examples of remarkable biologically significant discoveries. One such case revisited in this paper is the analysis of gastric cancer data [1]. The original paper used hierarchical clustering of both 7383 genes and 124 gastric cancer samples (malignant and pre-malignant). By inspecting the final "heat map" the authors observed a number of remarkable biclusters, which were then linked to various aspects of cancer etiology. However, the approach in [1] was heavily dependent on manual inspection to identify the final "biclusters". In particular, several sets of co-expressed genes were not grouped together by hierarchical clustering, and needed to be grouped manually by expert analysis. Moreover, it is difficult to assess whether such clusters are robust to any changes in the analysis, and whether different clustering attempts converge to a stable result. Consequently, there is a need for techniques that can guide such a process of discovering significant and worthwhile hypotheses for follow-up analysis.

This paper proposes such an exploratory technique. It is based on a form of biclustering [2], i.e., a method for automated discovery of highly correlated subsets of genes across a subset of samples, in combination with methods for evaluating the statistical significance and biological relevance of such biclusters. There are four main contributions that we make in this paper. First, we introduce a novel algorithm, called *bi-ordering*, which is in some respects a member of a family of biclustering techniques. This algorithm is benchmarked against several relevant biclustering algorithms in the literature [2][3][4][5]. Second, we introduce two novel statistical techniques for evaluating the significance of the generated groupings and orderings of multiple histological samples. Third, we assess the stability of the observed results by assessing the size of their "basin of attraction" as follows. In our experiments, random initializations of the algorithms yield hundreds of biclusters, which were then grouped into a manageable number of families of identical or very similar outcomes (called "super-biclusters") by a secondary phase of clustering the generated biclusters. The size of such a family is interpreted as the stability of the super-bicluster. We found that our technique can find a small set of highly stable super-biclusters, which correspond to distinct histopathological types in an existing gastric cancer data set [1]. Fourth, we demonstrate that the discovered super-biclusters have associated Gene Ontology (GO) terms with very significant p-values, which can serve as a basis for biological interpretation of the meaning of the associated gene modules.

2 The Bi-ordering Algorithm

We introduce a protocol for identifying and characterizing modules of genes that exhibit high statistical, biological and clinical significance. Our protocol, named *Bi-ordering Exploratory Analysis* (*BEA*), comprises six main stages as stated below:

1. Input: $n_G \times n_S$ gene expression data matrix of n_G genes for n_S samples.
2. Generate biclusters based on a bi-ordering of genes and samples.
3. Merge similar biclusters into "super-biclusters" to identify robust modules of co-expressed genes.
4. Annotate biclusters with histological and biological attributes to support their interpretation
5. Generate figures of merit (i.e., p-values) for:
    a. GO annotations,
    b. overrepresentation of histological categories in bicluster, and
    c. concordance of sample order with various phenotype gradients.
6. Develop biological interpretation of the results.

We briefly elaborate selected key stages of this protocol later in this section.

Bi-clustering – The term "biclustering", introduced by Cheng and Church in [2], refers to the identification of a sub-matrix with "significantly homogeneous entries". We have tested several existing biclustering algorithms, namely, Cheng and Church's algorithm [2] (*C&C*), SAMBA [3], biclustering by Gibbs sampling [5], and the ISA algorithm [4], which is closest to our algorithm. We have used open source implementations of these algorithms in our evaluation, i.e., SAMBA is tested using Expander [6], Gibbs sampling has been implemented by ourselves, and the biclustering toolbox BicAT [7] is used for the other two algorithms. We now introduce a novel algorithm pivotal in the generation of our results.

```
Algorithm 1 (Bi-Ordering Analysis - BOA)
1. Input: $n_G \times n_S$ data matrix $\left[x_{gs}\right]$, two cut-off thresholds $\theta_G$ and $\theta_S$.
2. Standardize data: first, for each gene (across all samples), to $std = 1$ and
   $median = 0$, then repeat this for each sample (across all genes).
3. Initialization: A non-empty subset of sample indices $S \subset \{1,...,n_S\}$.
4. Repeat the Steps a-d below until convergence (i.e., $G$ and $S$ stabilise):
    a. Update gene scores $f\,(g) \leftarrow \langle x_{gs} \rangle_{s \in S}$ for $g = 1,...,n_G$,
    b. Select genes: $G \leftarrow \left\{ g \;;\; f(g) - \langle f(g) \rangle_{g=1,...,n_G} > \theta_G \big/ \sqrt{|S|} \right\}$,
    c. Update sample scores $h\,(s) \leftarrow \langle x_{gs} \rangle_{g \in G}$ for $s = 1,...,n_S$,
    d. Select samples: $S \leftarrow \left\{ s \;;\; h(s) - \langle h(s) \rangle_{s=1,...,n_S} > \theta_S \big/ \sqrt{|G|} \right\}$.
5. Output: selected genes $G$, samples $S$ and ordering scores $f(g)$ & $h(s)$.
```

Here $\langle \cdot \rangle$ denotes an average, e.g., $\langle x_{gs} \rangle_{g \in G} := \sum_{g \in G} x_{gs} / |G|$ or $\langle x_{gs} \rangle_{s \in S} := \sum_{s \in S} x_{gs} / |S|$.

Note that the ordering scores *f* and *h* are uniquely determined by the selection of the bicluster (*G,S*). Other variants of BOA are possible, such as selecting significantly down-regulated genes or using *G* and *S* of fixed size. The attraction of the last option is that the algorithm is guaranteed to converge in such a case (formal proof not included), which in practice does not always happen for the previous two options.

Merging biclusters – In order to identify a robust set of biclusters, we run BOA with 1000 different initial subsets of samples, each drawn randomly with probability 0.2. For the thresholds θ_G and θ_S actually used in our experiments algorithm always converged. Some of generated biclusters were identical, while others were very similar to each other. We then applied a hierarchical clustering algorithm using complete linkage to group similar biclusters into *super-biclusters* (*SBC*). We have used the Jaccard coefficient on genes as a similarity measure between biclusters and a similarity threshold of 0.5 to generate super-biclusters. (A similar procedure could be applied to samples, though here we have focused on genes, which are the dominant and far more complex dimension to handle in this dataset.)

Ordering score – An important aspect of our analysis protocol is the ability to assign an ordering score to samples, *h*(*s*), and genes, *f*(*g*), for a given bicluster. The gene score *f* orders all genes according to the average expression level across samples in the bicluster.

2.1 Figures of merit

Saturation Statistics - The homogeneity of samples in a bicluster can be evaluated if we are given a prior classification of each sample (e.g., its cancer subtype) as a label. Ideally, each bicluster should be dominated by one or more similar classes. Thus, we can use the p-value of the hyper-geometric distribution to evaluate the purity of biclusters according to the classification of samples. A similar evaluation was applied to the single most abundant class within a bicluster in [3]. However, if some genes are co-regulated across *multiple* classes, calculating p-values on a single class is not an adequate representation of accuracy. To address this limitation, we introduce a generalized approach where significance is calculated for the group of classes with the best p-value. The *single-class saturation* and *multiple-class saturation* are called SCS and MCS, respectively.

Gene Ontology (GO) annotation – Given that each gene's expression in a bicluster is highly similar with respect to other genes in the cluster, it is expected that the collection of genes as a whole are likely to be involved in a similar biological process. In order to determine this, the structured vocabulary of the Gene Ontology [8] (GO) was used to help uncover the biological processes represented by each of the SBCs. As each gene can be annotated to one or more terms within the GO, we can determine which GO terms are statistically overrepresented within a group of genes. We used GOSTAT [9] to determine the statistically overrepresented terms within each SBC for the biological process branch of the GO.

Trend statistics – Another method to evaluate the significance of a super-bicluster is to compare the ordering of samples *h(s)* generated by the super-bicluster with any relevant ordering *y(s)* of the samples based on their biology, e.g., the progression of the cancer in the sample. We can test the agreement of samples ordered according to *h*(*s*) with this progression *y(s)*. We use the following extension of the Mann-Whitney statistics, $U := \left|\{(s,s')\ ;\ \ h(s) < h(s')\ \&\ y(s) < y(s')\}\right|$, for this purpose. For random scoring h (our H_0 hypothesis) this random variable has an approximately normal distribution with mean $\sum N_i / 2$ (where N_i denotes the size of the label classes) and variance $\sum_{i<j} N_i N_j \left(N_i + N_j + 1\right)/12 + \sum_{i<j<j'} N_i N_j N_{j'} / 6$. Note that other tests for trend statistics can also be used here, such as Jonckheere-Terpstra or Page's test.

3 Experimental Evaluation

In this section, we analyze the performance of our algorithm on a real gene expression dataset for gastric cancer [1]. The main reason for this choice is the availability of local expertise in the biology of this disease. We compare the performance of our algorithm to the results obtained from the algorithms in [2,3,4,5] by using the parameter settings recommended in those papers or by observing the best results obtained under different parameter settings. Similar results were obtained on lymphoma data [10] but are omitted in this abstract due to space limits.

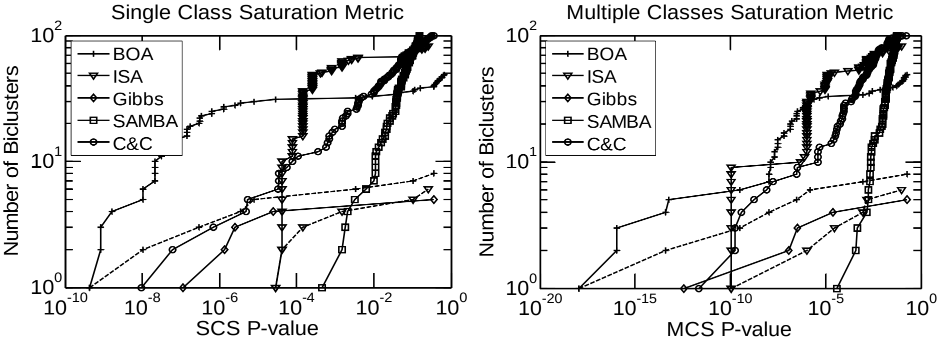


Fig. 1. Gastric cancer benchmark results for five bi-clustering algorithms. We plot the number of unique biclusters (*continuous lines*) and super-biclusters (*dotted lines*) with a *p*-value below the threshold indicated by the *x*-axis. We have used the SCS (*left sub-figure*) and MCS (*right sub-figure*) metrics to calculate the p-values. We have applied 1000 random initializations for BOA and ISA.

After applying the gene filtering as described in [1], we have n_G = 7383 gene expressions evaluated for n_S = 124 human tissue samples. Excluding two singletons, there are 6 different phenotypes in the data, of which three are subtypes of gastric cancer (35 Diffuse, *DGC*; 22 Intestinal, *IGC*; 7 Mixed, *MGC*) and the other three are pre-malignant conditions: 26 chronic gastritis (*CG*), 22 intestinal metaplasia (*IM*) and 10 "*normal*", i.e., non-inflamed mucosa tissue removed during surgery for the gastric cancer. Now we briefly discuss the algorithmic aspects and setup of the experiment. The biological relevance will be discussed in the following section.

3.1 Specific BOA Settings and Performance for the Gastric Cancer Experiment

We have evaluated 10 different pairs of settings for thresholds $\theta_G \in \{4, 4.5, ..., 6\}$ and $\theta_S \in \{3, 3.5, ..., 5.5\}$. For those we have we found that the minimal p-values ranged between 4.3×10^{-10} and 1.6×10^{-9} (with 9 of the 11 achieving the minimum) for the SCS metric, and 3.4×10^{-27} and 4.0×10^{-14} for the MCS metric. For further analysis we have chosen a mid-range pair $\theta_G = 5$ and $\theta_S = 4.5$ for which, additionally, all 1000 initialisations of BOA converged.

3.2 Biological Analysis of BOA Results

In this section, we focus on validating the biological significance of our findings for the gastric cancer dataset by comparing the biclusters with those reported in a previous study. In [1], hierarchical clustering was applied to the gastric cancer (cDNA) data set and several regions of genes related to different cancer types or pre-malignant states were annotated (labelled A – K in [1, Figures 1-2]). To validate our biclusters, we determined the intersection between genes in these regions identified and the genes appearing in the prototypes of the eight super-bi-clusters (SBC_1 – SBC_8) generated by the BOA algorithm. The results are shown in Table 1. Note that the two largest super-biclusters (SBC_6 and SBC_7) were a close match for the two most prominent biclusters in [1] (regions B & K). Moreover, the super-bicluster SBC_2 linked two separated but related biclusters in [1] (regions E & F), while the regions D1 to D3 that needed to be manually grouped in [1] were automatically grouped by our method in SBC_5.

We then considered the significance of these super-biclusters in terms of the three types of figures of merit discussed in Section 2.1, namely, the MSC p-values, the p-value of the most significant GO annotation, and the p-value of the correlation of the order according to *h*(*s*) with the "*Malignancy Score*" *y*(*s*) defined as follows. First, following advice from experts, to each sample *s* we allocate *y*(*s*) = 1 for the *phenotype* =Normal, 2 for =CG, 3 for =IM and finally 4 for any gastric cancer (DGC, IGC or MGC sample). We then tested the significance of the agreement of the samples ordered according to the *h*(*s*) score generated by the BOA algorithm with trend *y*(*s*) (see Section 2.1). Table 2 shows that *h* for SBC_5 and SBC_7 and to a lesser extent SBC_3 are very significantly correlated with *y*. The heat map of SBC_7 (Figure 2) shows that the ordering induced by the bicluster has a clear (negative) correlation with the Malignancy Score of the samples.

Note that the sample scores *h*(*s*) have a plausible biological interpretation. As the genes in the BOA bi-cluster are approximately uniformly over-expressed, *h*(*s*) is a measure of their average over-expression, and so, of over-expression of the GO annotations linked with the SBC. For instance, the highly significant Malignancy Score for the SBC_7 (Table 2) indicates that the process of "generation of precursor

Table 1. Overlaping genes between prototypes for super biclusters and functional regions in [1]. In the second row we show the number of genes in the SBC prototype.

Region in [Bou03]			SBC_1	SBC_2	SBC_3	SBC_4	SBC_5	SBC_6	SBC_7	SBC_8
Symbol	Annotation	No.Genes	41	217	194	158	227	409	515	146
B	Mitochondrial	665	0	0	0	0	0	1	416	9
D1-D3	Proliferation	201	0	0	0	0	76	0	0	0
E	Intestinal	294	1	81	0	0	0	0	1	44
F	Intestinal	157	0	112	0	0	7	1	0	27
G	Squamous	37	25	0	0	0	0	0	0	0
H	Inflamation	330	7	0	117	135	9	7	0	30
K	Extracellular	877	3	0	67	0	74	392	1	0

Table 2. Numerical characterisations and biological relevance of super biclusters generated by BOA. Note that the negative sign, `-`, in Malignancy Score for SBC_7 and SBC_8 indicates the significance of agreement with the reverse order. In the second column of the table, the numbers of biclusters that converged to a particular super-bicluster are given.

	# biclusters		p-value			Most significant annotation
SBC	SBC	Proto-type	MCS	Malignancy Score	GO (most sign.)	GO
SBC_1	11	6	9.4E-04	1.8E-13	5.1E-09	epidermis development
SBC_2	188	7	1.0E-08		7.1E-07	lipid metabolic process
SBC_3	2	1	1.5E-06	5.5E-08	3.2E-32	immune system process
SBC_4	96	2	1.8E-01		2.0E-53	immune system process
SBC_5	15	15	1.1E-18	7.7E-21	1.8E-14	cell cycle process
SBC_6	328	11	3.0E-07	4.9E-08	1.8E-20	multicellular organismal process
SBC_7	359	229	4.0E-14	-5.4E-22	3.2E-22	gen. of precursor metab. & energy
SBC_8	1	1	3.0E-10	-5.2E-08	2.2E-02	lipid metabolic process

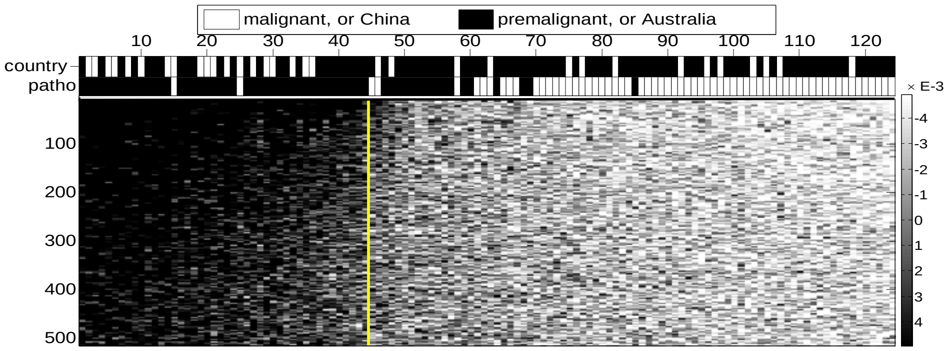


Fig. 2. Heat map for the most prominent super-bicluster, SBC_7, generated by the BOA algorithm for the gastric cancer data. We observe the strong gradation from least "malignant" normal samples, though CG and IM, to the malignant samples (combined intestinal, diffuse and mixed gastric cancers). The probability of obtaining such or better ordering by random chance was estimated as < 5.4×10^{-22} using the trend statistics, Section 2.1. The vertical axis shows the 515 most significant genes, while the horizontal axis shows the final order of samples generated by the BOA algorithm. The white vertical line indicates the right boundary of samples in the bicluster.

metabolism & energy” is over active in pre-malignant samples relatively to the malignant cases. On the over hand, the results for SBC_5 strongly indicate that “cell cycle process” is significantly more active in the malignant samples.

The generated results including the GO and clinical correlations were the basis of an evaluation by expert biologists and clinicians who judged that the formal data processing protocols as discussed generated a number of significant biological hypotheses warranting follow-up wet lab experiments. In summary, the BOA results have shed new light on preexisting themes in gastric cancer etiology. The resulting bi-orderings represent successive steps in cancer progression and distinct histopathological types of the disease. Specifically, SBC_1 represents epithelial morphology, typical to squamous samples; SBC_2 and SBC_8 are typical intestinal lipid metabolism signatures, observed in intestinal metaplasia premalignant samples; SBC_3 and SBC_4 represent a novel split of the inflammatory signature that in [1] were merged as one signature; SBC_5 represents the proliferation signature described in [1] for intestinal type gastric cancer; SBC_6 reflects the extracellular matrix deposition typical to diffuse type cancer, and elevated in all cancer samples compared to premalignant samples; SBC_7 represents the metabolic stress observed in chronic gastritis samples, possibly due to elevated H. Pylori infection. A more detailed discussion will be included into the full version of the paper.

3.3 Brief comparison to ISA and Gibbs Algorithms

The BOA algorithm is very similar to ISA. However, the main objective of ISA is discerning “co-regulated’’ gene modules, while the association with phenotype classes (conditions) is not important, whereas it is of prime interest for our medical application. The main formal differences resulting in different performance are: (i) ISA starts with an initialisation of a subset of genes; (ii) the two sided test is used for the selection of samples; (iii) samples are weighted, with possibly negative weights, so different conditions, say with up-regulated and down-regulated genes, can be joined in the same bi-cluster. Consequently, ISA aims at generating “constant column’’ biclusters while BOA’s objective is the “constant’’ bicluster [11]. Figure 2 shows that BOA generates more significant biclusters in terms of SCS and MCS.

The evaluation of GO annotation for both ISA and Gibbs shows that they are capable of generating biclusters of significance comparable to BOA. These algorithms generated 6 and 5 SBCs, respectively, with similar gene sets to the SBCs of BOA. For example, the GO annotations “generation of precursor metabolites and energy“ and “oxidative phosphorylation’’ significantly associated with SBC7 of BOA (*p*-value are 3e-22 and 4e-18, see Table 2) are also found by the ISA algorithm (*p*-value 3e-8 and 4e-6) and Gibbs algorithm (*p*-value 1e-30 and 5e-13). Similarly, the “multicellular organismal process’’ and “multicellular organismal development’’ annotations (significant for diffuse-type gastric cancer) in SBC6 of BOA, were also found by the ISA and Gibbs algorithms. However, we have observed that the BOA algorithm usually has better performance than either ISA of Gibbs in terms of trend statistics, in particular, the evaluation of malignant progression (Section 3.2).

Conclusions

In this paper we have presented a novel method of bi-ordering genes and samples from microarray data, together with two novel statistical techniques for evaluating the significance of the generated groupings and orderings of multiple histological samples. In comparison to several existing algorithms in the literature, our method is able to generate highly robust and statistically significant gene modules with respect to sample histological annotations on a gastric cancer dataset. The results of our analysis closely match reported theories of gastric cancer tumorgenesis, and have helped to identify promising hypotheses for further investigation in cancer research. We also show that other biclustering algorithms can be utilized as a basis of exploratory bi-ordering analysis of genomic data.

Acknowledgments. National ICT Australia (NICTA) is funded by the Australian Government's Department of Communications, Information Technology and the Arts and the Australian Council through Backing Australia's Ability and the ICT Center of Excellence program.

References

1. A. Boussioutas, et al., Distinctive Patterns of Gene Expression in Premalignant Gastric Mucosa and Gastric Cancer, Cancer Research, 63, pp. 2569–2577, 2003.
2. Y. Cheng and G. M. Church, Biclustering of Expression Data, *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, 2000.
3. A. Tanay, R. Sharan and R. Shamir, Discovering Statistically Significant Biclusters in Gene Expression Data, *Bioinformatics,* Vol. 18, pp. S136-S144, 2002.
4. J. Ihmels, S. Bergmann and N. Barkai, Defining transcription modules using large-scale gene expression data, *Bioinformatics*, Vol. 20, No. 13, pp. 1993-2003, 2004.
5. Q. Sheng, Y. Moreau, and B. De Moor, Biclustering Microarray Data by Gibbs Sampling *Bioinformatics,* Vol. 19, pp. ii196-ii205, 2003.
6. R. Sharan, et al., CLICK and EXPANDER: A System for Clustering and Visualizing Gene Expression Data, *Bioinformatics*, Vol. 19, No. 14, pp. 1787-1799, 2003.
7. S. Barkow, et al., BicAT: a biclustering analysis toolbox, *Bioinformatics*, Vol. 20 No. 10, pp. 1282-1283, 2006.
8. M. Ashburner, et al., Gene Ontology: tool for the unification of biology, *Nat Genet*, Vol. 25, pp. 25-29, 2000.
9. T. Beissbarth and T.P. Speed, GOstat: find statistically overrepresented gene ontologies within a group of genes, *Bioinformatics*, Vol. 20, pp. 1464-1465, 2004.
10. A. Alizadeh et al., Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling, *Nature* Vol. 403, pp. 503-511, 2000.
11. S. C. Madeira, A. L. Oliveira, Biclustering algorithms for biological data analysis: a survey, *IEEE/ACM Trans. Comput. Biol. Bioinformatics* (2004), pp.ii196-ii205, 2003.

Raw genotypes vs haplotype blocks for genome wide association studies by random forests

Vincent Botta[1,3], Sarah Hansoul[2,3], Pierre Geurts[1,3], and Louis Wehenkel[1,3]

[1] Department of Electrical Engineering and Computer Science
[2] Animal Genomics
[3] GIGA-Research, University of Liège, B4000 Belgium

Abstract. We consider two different representations of the input data for genome-wide association studies using random forests, namely raw genotypes described by a few thousand to a few hundred thousand discrete variables each one describing a single nucleotide polymorphism, and haplotype block contents, represented by the combinations of about 10 to 100 adjacent and correlated genotypes. We adapt random forests to exploit haplotype blocks, and compare this with the use of raw genotypes, in terms of predictive power and localization of causal mutations, by using simulated datasets with one or two interacting effects.

Key words: Random forests, genome-wide association studies, complex diseases, variable importance measures

1 Introduction

The majority of important medical disorders (f.i. susceptibility to cancer, cardiovascular diseases, diabetes, Crohn's disease) are said to be complex. This means that these diseases are influenced by multiple, possibly interacting environmental and genetic risk factors. The fact that individuals differ in terms of exposure to environmental as well as genetic factors explains the observed inter-individual variation in disease outcome (i.e. phenotype). The proportion of the phenotypic variance that is due to genetic factors (heritability) typically ranges from less than 10 to over 60 % for the traits of interest. The identification of genes influencing susceptibility to complex traits reveals novel targets for drug development, and allows for the implementation of strategies towards personalized medicine.

Recent advances in marker genotyping technology allow for the genotyping of hundreds of thousands of Single Nucleotide Polymorphisms (SNP) per individual at less than 0.1 eurocents per genotype. The identification of genomic regions (i.e. loci) that influence susceptibility to a given disease can now be obtained by means of so-called "genome-wide association studies" (GWAS). Basically, the idea behind GWAS of complex diseases is to genotype a collection of affected (cases) and unaffected (controls) individuals for a very large number of genetic markers spread over the entire genome. Typically, one disposes of a cohort of a few hundred to a few thousand individuals, a fraction of them (typically about 50%) having a certain phenotype (e.g. disease status, or treatment response status), and the rest of them being controls (individuals representative of the genetic variation in the studied population and who do not present the studied phenotype). In this domain, supervised learning, and in particular Random Forests, has been recently proposed to circumvent the limitations of standard approaches based on univariate statistical tests [1–3].

In this paper, we study two different representations of the input data for the application of supervised learning in GWAS, namely the raw SNP genotypes on the one hand, and on the other hand new features derived from groups of strongly correlated SNPs (i.e. the haplotype blocks; those blocks are transmitted from parents to offspring during the recombination of parental chromosomes). We propose an adaptation of Random Forests to handle haplotype blocks as well as SNPs. Currently, available real-life datasets are still being investigated by the geneticists, for this reason, as a first step, we compare the two approaches empirically on simulated datasets with one or two independent or interacting causal mutations. Our two contributions with respect to previous work are the exploitation of haplotype blocks and its systematic evaluation on high density simulated datasets, both for genetic risk assessment and for the localization of causal mutations.

The rest of the paper is organized as follows. In Section 2, we describe the algorithms, while Section 3 presents the simulated datasets and simulation results. We conclude in Section 4 with discussions and future work directions.

2 Methods and algorithms

2.1 Random forests

From a machine learning point of view, a GWAS of a complex disease is a binary classification problem, with a very large number of raw variables, each one corresponding to a different SNP and having only three possible values (homozygous wild, heterozygous and homozygous mutant). On top of this very high p/n ratio, these problems are also generally highly noisy, and the raw input variables are strongly correlated (due to linkage disequilibrium).

The nature of the problem puts several constraints on candidate supervised learning methods. The method needs to find a small number of relevant variables among a very large number of irrelevant ones, and thus incorporate some feature selection mechanism. It needs to be sufficiently expressive to take into account possible interactions between SNPs. Computationally, the algorithm should furthermore be able to cope with hundreds of thousands of variables and thousands of individuals. Tree-based ensemble methods provide a good tradeoff along these criteria. Among existing ensemble methods, we focus in this paper on the Random Forests algorithm [4]. This algorithm grows each tree of the ensemble from a bootstrap sample drawn from the original data, using the CART algorithm (without pruning) with a modified node splitting procedure. At each test node, the algorithm selects the best split using the standard CART procedure but from a subset of only K attributes selected at random among all candidate attributes.

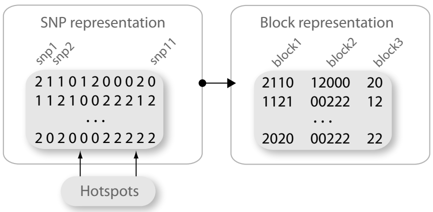


Fig. 1. Database transformation from SNPs to haplotype blocks.

The algorithm performances depend on the number T of trees in the ensemble (the larger the better) and on the number K of selected attributes at each test node, whose optimal value is problem dependent.

2.2 Individual SNP and haplotype block representations

Figure 1 shows the two representations of input data that we will use for growing Random Forests, and how the block contents are computed from the SNPs. SNPs are arranged as they appear along the chromosome and the integer values $\{0, 1, 2\}$ represent the number of mutant alleles at the corresponding position.

In order to apply Random Forests on the raw genotype data, we merely consider each SNP as a numerical variable. To handle attributes representing the contents of haplotype blocks, we propose the following adaptation of the node-splitting procedure:

- At each test-node, K blocks are selected at random.
- For each block b, we proceed as follows:
  - for each SNP i in b, we compute from the subset of *cases* (resp. *controls*) at the test-node the frequency of its three possible values $(f^b_{i,j})_{case}$ (resp. $(f^b_{i,j})_{control}$) $(i = 1, \ldots, l_b, j = 0, 1, 2)$, where l_b denotes the number of SNPs in b;
  - for each case or control x, we compute the two probabilities :

    $$P(x|case, b, node) = \prod_{i=1}^{l_b} (f^b_{j,s_i(b,x)})_{case} \tag{1}$$

    and

    $$P(x|control, b, node) = \prod_{i=1}^{l_b} (f^b_{j,s_i(b,x)})_{control}, \tag{2}$$

    where $s_i(b, x)$ denotes the value of the ith SNP of b for this individual x;[4]

[4] This is a maximum likelihood based estimation of the conditional probability that the observed haplotype is drawn from the population of cases (resp. controls) reaching the current node, assuming class conditional independance of the SNPs in the block b.

- then, an optimal cutoff is determined on the probability ratio:

$$\frac{P(x|case, b, node)}{P(x|control, b, node)} \tag{3}$$

using the standard CART procedure for numerical variables.

– The best split among the K optimal splits is selected to split the node.

Notice that the motivation behind the block-wise approach is to reduce the number of features by grouping correlated SNPs, and thus to improve the robustness of the method. In our description, we left open the question of the determination of the blocks. In our experiments, we will compare two approaches. First, haplotype blocks delimited by *HapMap* hotspot list generated from a panel of 5 populations from which our simulated data will be derived, second, haplotype blocks reconstructed from a linkage disequilibrium map computed by the *Haploview* software [5] applied on our simulated datasets.

2.3 Localization of causal mutations

Several importance measures have been proposed in the literature to derive from a tree ensemble a ranking of candidate attributes according to their relevance for predicting the output. In the context of GWAS, such measures may be used to identify the SNPs or haplotype blocks closest to the causal mutation loci. In our simulations we use to this end the information theoretic measure proposed in [6] computing for each attribute the total reduction of class entropy (the sum over all test-nodes of the ensemble where this attribute is used, of the local reduction in entropy weighted by the local sample size).

3 Experiments

3.1 Simulated dataset

We used *gs* [7] to generate samples based on *HapMap* data [8] with linkage disequilibrium patterns similar to those in actual human populations. We focus our experiments on chromosome 5 (because its size is close to the mean size of other human chromosomes). The raw input variables were obtained by taking SNPs spaced by 10 kilobases from the *HapMap* pool to reproduce classical GWAS conditions, and the causal disease loci were removed from the input variables.

Five different disease models were tested: two models with one disease locus, and three models with two interacting loci. Tables 1 and 2 give the penetrance matrix for each model. These tables report the probabilities of being affected for each possible genotype of the locus or loci. Lower case letters (a, b) denote wild alleles and upper case letters (A, B) denote mutant alleles. We introduce a noise level of 0.005 to simulate environmental effects. The 3 two loci models were selected among the most common disease models referenced in [9].

Table 1. The one-locus disease models that were investigated in this study.

1A			1aA		
aa(0)	**aA**(1)	**AA**(2)	**aa**(0)	**aA**(1)	**AA**(2)
0.005	0.005	0.100	0.005	0.100	0.250

Table 2. The two-locus disease models that were investigated in this study.

	2DD			2RD			2XOR		
	bb(0)	**bB**(1)	**BB**(2)	**bb**(0)	**bB**(1)	**BB**(2)	**bb**(0)	**bB**(1)	**BB**(2)
aa(0)	0.005	0.005	0.005	0.005	0.005	0.005	0.005	0.005	0.100
aA(1)	0.005	0.100	0.100	0.005	0.005	0.005	0.005	0.005	0.100
AA(2)	0.005	0.100	0.100	0.005	0.100	0.100	0.100	0.100	0.005

The first two disease models (Table 1) contain one susceptibility locus. In the first one, 1A, two copies of the mutant allele increase the risk of being affected. The second one 1aA is additive: the risk increases with the number of mutant alleles present at the susceptibility locus. The three disease models described in Table 2 involve two susceptibility loci. For the 2DD model (dominant-dominant), the two loci are dominant, meaning that at least one copy of the mutant allele at the two loci is required for the risk to increase. The 2RD model (recessive-dominant) requires two copies of disease alleles from the first locus and at least one disease allele from the second. Finally, in the 2XOR model, two mutant copies at one locus or three mutant copies at any of them increase the disease risk.

In the first (raw) data representation, the different databases are composed of about 14000 numerical variables. This number was reduced to 2000 variables of *HapMap* blocks and 6500 variables of blocks obtained with *Haploview*.

3.2 Protocol

For each disease model, we generated 7000 individuals (with 50% of cases) that we divided into 2000 individuals for learning and 5000 individuals for testing. The learning sample was divided into 4 subsets of size 500. A model was produced for each subset of size 500. We report average results (and standard deviations) over all subsets.

The predictive power was assessed using the area under the ROC curves (AUC) computed on the 5000 test samples and averaged over the training set and compared to the AUC obtained on the test samples with the Bayes optimal model deduced directly from the selected disease model. The latter is denoted as "Ref AUC" in the tables and figures reported below.

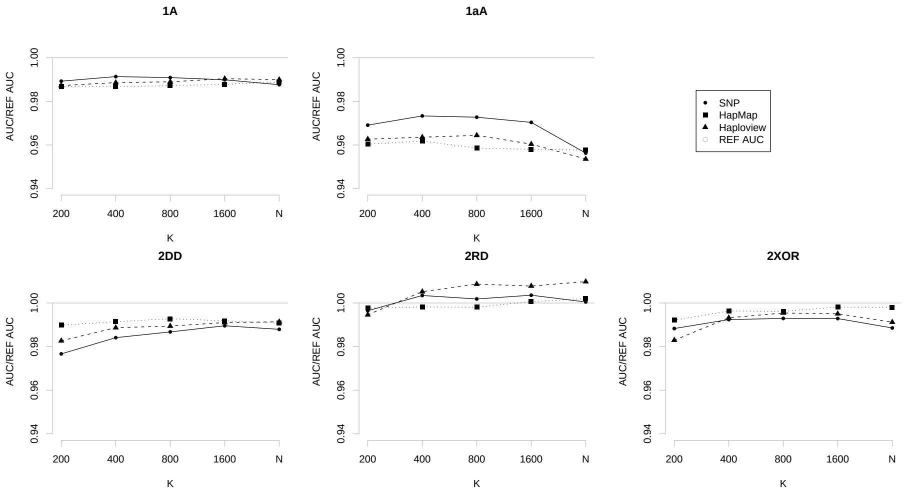


Fig. 2. Influence of parameter K on the five disease models. In plain: SNP; in dotted: *HapMap*; in dashed: *Haploview*; in gray: the ratio $\frac{AUC}{REF\,AUC}$. N denotes the total number of candidate attributes for each type of representation.

3.3 Empirical results

Parameter sensitivity. We first carried out some preliminary experiments to see the effect of the two parameters of the Random Forests algorithm. Given the important number of attributes, we observed that a quite large number of trees is necessary for the error to converge. In all our experiments, we therefore conservatively fixed T to 2000 trees. We also observed that small values of K (< 200) yield to suboptimal AUC values, and we therefore only explored higher values of K.

Figure 2 shows the evolution of the AUC for the five disease models with the K parameter and all three approaches (RF with raw SNPs, *HapMap* and *Haploview* blocks). Note that in this graph, N is very different from one method to another (resp. 14000, 2000, and 6500 for SNP, *HapMap* and *Haploview*). We observe that *HapMap* and *Haploview* produce slightly better results than SNPs for the models 1A, 2DD, 2RD, and 2XOR. Typically, larger values of K yield very close to optimal results. Note however that the maximal AUC is usually already obtained with significantly lower values of K (1600), which correspond also to smaller computational requirements. In our experiments below, we will thus present only the results for this setting.

Table 3. AUC (average ± std. dev.) for K=1600.

	SNP	HapMap	Haploview	Ref AUC
1A	0.7311 ± 0.0048	0.7296 ± 0.0061	**0.7315** ± 0.0048	0.7386
1aA	**0.7901** ± 0.0016	0.7800 ± 0.0025	0.7820 ± 0.0020	0.8142
2DD	0.8112 ± 0.0012	**0.8131** ± 0.0012	0.8124 ± 0.0011	0.8198
2RD	0.6377 ± 0.0072	0.6358 ± 0.0044	**0.6403** ± 0.0033	0.6354
2XOR	0.7927 ± 0.0037	**0.7969** ± 0.0040	0.7944 ± 0.0012	0.7984

Predictive power. Table 3 reports AUCs for the different methods for all considered disease models with $K = 1600$. Overall, the results of the SNP representation and the two types of blocks are very close to each other and to the "Ref AUC" on most of the models. Blocks outperform the SNPs on 1A, 2DD, 2RD and 2XOR. The *HapMap* blocks outperform the *Haploview* on 1A, 2DD and 2XOR.

Localisation of causal mutations. For the one locus model the causal mutation (A) is located at position 1599; for the two-loci model the first causal mutation (A) is located also at position 1599, while the second one (B) is located far away, at position 11175. Figure 3 shows the SNP importances over the chromosome 5, while Figure 4 provides a zoom of the variable importances of the three methods over the regions close to the two causal loci of the two-loci disease models. We observe that in all cases, except for 2RD, the genomic regions containing the two causal mutations are very well localized.

4 Conclusions

The preliminary results obtained in this paper show promising perspectives. In particular, the different methods obtain rather good AUCs as compared with the theoretical upper bound derived from the disease models. The different methods are also able to predict and to localize the disease loci, rather well. We observed that most often our adaptation of Random Forests to the block representation of the data provides marginally superior results in terms of risk prediction than their direct application to the raw genotype data. Results not reported in this paper with different ensembles of trees do not contradict these findings.

An interesting direction of future research will be the refinement of the treatement of the haplotype block structure in supervised learning. In the context of tree-based methods, we envisage two extensions of the splitting procedure: first, there are various possible ways to improve the way likelihoods are computed within a block, e.g. by relaxing class-conditional independance; second, one could use overlapping (and of randomized length) block structures or greedily search for optimal block size around a SNP of interest locally at each tree node, instead of exploiting an a priori fixed block structure. More generally, we believe that the simultaneous exploitation of blocks and SNPs may also be of interest.

Future work will also consider more complex disease models, real-life datasets, quantitative traits, as well as even higher density genotyping, in the limit towards the next generation of full genomic resequencing based genotyping.

Acknowledgments

This paper presents research results of the Belgian Network BIOMAGNET (Bioinformatics and Modeling: from Genomes to Networks), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. VB is recipient of a F.R.I.A. fellowship and SH and PG are respectively postdoctoral research fellow and Research Associate of the F.R.S.-FNRS. The scientific responsibility rests with the authors.

References

1. Costello, T., Swartz, M., Sabripour, M., Gu, X., Sharma, R., Etzel, C.: Use of tree-based models to identify subgroups and increase power to detect linkage to cardiovascular disease traits. BMC Genetics **4**(Suppl 1) (2003) S66
2. Bureau, A., Dupuis, J., Hayward, B., Falls, K., Van Eerdewegh, P.: Mapping complex traits using random forests. BMC Genetics **4**(Suppl 1) (2003) S64
3. Lunetta, K., Hayward, L.B., Segal, J., Van Eerdewegh, P.: Screening large-scale association study data: exploiting interactions using random forests. BMC Genetics **5**(1) (2004) 32
4. Breiman, L.: Random forests. Machine Learning **45**(1) (2001) 5–32
5. Barrett, J.C., Fry, B., Maller, J., Daly, M.J.: Haploview: analysis and visualization of ld and haplotype maps. Bioinformatics **21**(2) (2005) 263–265
6. Wehenkel, L.: Automatic learning techniques in power systems. Kluwer Academic, Boston (1998)
7. Li, J., Chen, Y.: Generating samples for association studies based on hapmap data. BMC Bioinformatics **9** (2008) 44
8. Consortium, T.I.H.: The international hapmap project. Nature **426**(6968) (Dec 2003) 789–796
9. Li, W., Reich, J.: A complete enumeration and classification of two-locus disease models. Hum Hered **50**(6) (2000) 334–349

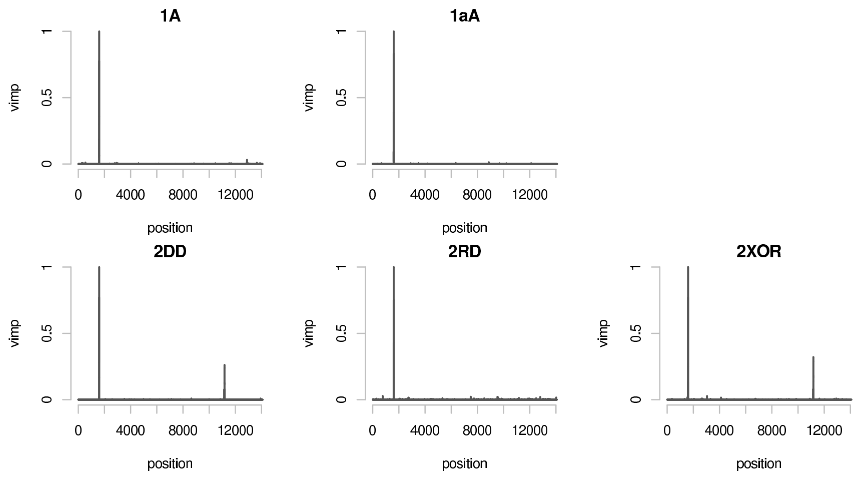


Fig. 3. Variable importances with SNP: overview over chromosome 5 (average normalized values over 4 learning samples), $K = 1600$, $T = 2000$.

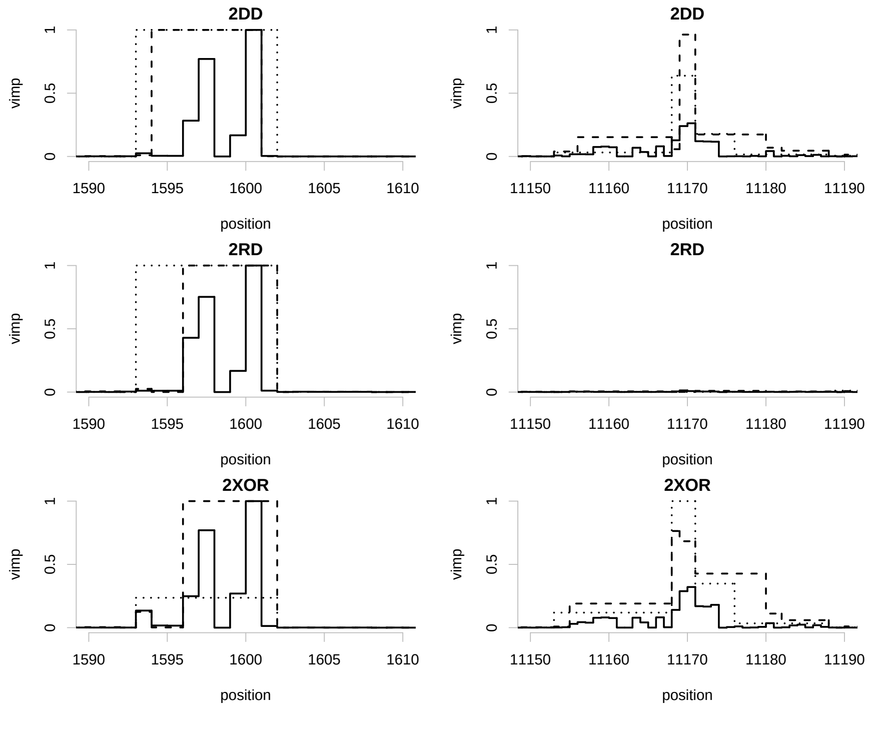


Fig. 4. Variable importances: zoom around the causal loci. In plain, the SNP, in dotted, the *HapMap* blocks, and in dashed, the *Haploview* blocks. Learning sample size of 500 (average values over 4 learning samples), $K = 1600$, $T = 2000$.

Poster abstracts

List of posters

BysCyc: A Bayesian Logic for the Integrative Analysis of Knowledge and Data in Genetic Association Studies

P. Antal, A. Millinghoffer, G. Hullám, Cs. Szalai, and A. Falus

Dept. of Measurement and Information Systems
Budapest University of Technology and Economics
Inflammation Biology and Immunogenomics Research Group
Hungarian Academy of Sciences
Dept. of Genetics, Cell- and Immunobiology
Semmelweis University, Hungary

Despite the grand promises of the postgenomic era, such as personalized prevention, diagnosis, drugs, and treatments, the landscape of biomedicine looks more and more complex, particularly in genetic association studies (GAS) (e.g., see [16, 15, 20, 8]). A promising trend in data analysis is the Bayesian approach, which is able to cope with small sample size, fusion of data and knowledge, challenges of multiple testing, meta-analysis, and positive results bias [24, 3, 18, 26, 25, 12]. Within this approach we introduced the methodology of the Bayesian Multilevel Analysis of the relevance of input variables [1, 2]. It uses Bayesian networks [9] for the analysis of relevance at the levels of Markov Blanket Memberships, Markov Blanket sets, and Markov Blanket graphs, which correspond to the pairwise, multivariate, and the multivariate-interactionist levels.

Beside data analysis another bottleneck in GAS research is the interpretation of the results. There is a lot of additional information for the molecular "grounding" of effects of genetic variations both w.r.t. genetic regulation and in protein functions (e.g., see [11, 4, 14, 17, 10, 7]), and its their integration is challenging, particularly the integration of the structural knowledge or the networks of mechanisms [19, 21, 22]. These methods, e.g. the Ingenuity System [22], allow for the fusion of literature, expert knowledge, and the results of statistical data using a (1) pairwise (univariate), (2) diagrammatic (propositional), and (3) deterministic framework.

We developed and implemented a general Bayesian method, which supports such a fusion at the multivariate level with interactions exploiting the power of probabilistic first-order logic [13]. In this poster we present the proposed Bayesian logic, enumerate the main types of inference, and illustrate typical queries in GAS.

Finally, we present links to recent, related developments in the field of probabilistic world-wide web and probabilistic databases [5, 6]. The genomics of asthma will serve as an application domain [23].

References

1. P. Antal, A. Millinghoffer, G. Hullám, G.Hajós, Cs. Szalai, and A. Falus. A bioinformatic platform for a Bayesian, multiphased, multilevel analysis in immunogenomics. In D.R.Flower M.N.Davies, S.Ranganathan, editor, *Bioinformatics for Immunomics*, pages –. Springer, 2008.
2. P. Antal, A. Millinghoffer, G. Hullám, Cs. Szalai, and A. Falus. A bayesian view of challenges in feature selection: Feature aggregation, multiple targets, redundancy and interaction. *JMLR Proceeding*, 42:–, 2008.
3. D. J. Balding. A tutorial on statistical methods for population association studies. *Nature*, 7:781–91, 2006.
4. Worth CL, Bickerton GR, Schreyer A, Forman JR, Cheng TM, Lee S, Gong S, Burke DF, and Blundell TL. A structural bioinformatics approach to the analysis of nonsynonymous single nucleotide polymorphisms (nssnps) and their relation to disease. *J Bioinform Comput Biol.*, 5(6):1297–318, 2007.
5. Paulo Cesar G. da Costa, Kathryn B. Laskey, , and Kenneth J. Laskey. Pr-owl: A bayesian ontology language for the semantic web. In -, 2007.
6. Nilesh Dalvi and Dan Suciu. Efficient query evaluation on probabilistic databases. In *Proceedings of the 30th VLDB Conference*, pages –, 2004.
7. Richard J Dobson, Patricia B Munroe, Mark J Caulfield, and Mansoor AS Saqi. Predicting deleterious nssnps: an analysis of sequence and structural attributes. *BMC Bioinformatics*, 7:217, 2006.
8. W. Gregory Feero, Alan E. Guttmacher, and Francis S. Collins. The genome gets personal-almost. *JAMA*, 299(11):1351–2, 2008.
9. N. Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, 303(5659):799 – 805, 2004.
10. Byoung-Chul Kim, Woo-Yeon Kim, Daeui Park, Won-Hyong Chung, Kwang sik Shin, and Jong Bhak. Snp@promoter: a database of human snps (single nucleotide polymorphisms) within the putative promoter regions. *BMC Bioinformatics*, 9, 2008.
11. Bao L, Zhou M, and Cui Y. nssnpanalyzer: identifying disease-associated nonsynonymous single nucleotide polymorphisms. *Nucleic Acids Res.*, 33:W480–2, 2005.
12. Harold P. Lehmann and Steven N. Goodman. Bayesian communication, a clinically significant paradigm for electronic publication. *Journal of the American Medical Informatics Association*, 7:254–266, 2000.
13. A. Millinghoffer, G. Hullám, and P. Antal. On inferring the most probable sentences in bayesian logic. In *Workshop notes on Intelligent Data Analysis in bioMedicine And Pharmacology (IDAMAP-2007)*, pages 13–18, 2007.
14. Wang P, Dai M, Xuan W, McEachin RC, Jackson AU, Scott LJ, Athey B, Watson SJ, and Meng F. Snp function portal: a web database for exploring the function implication of snp alleles. *Bioinformatics*, 22(14):e523–9, 2006.
15. Timothy R. Rebbeck, Muin J. Khoury, and John D. Potter. Genetic association studies of cancer: Where do we go from here? *Cancer Epidemiology Biomarkers & Prevention*, 16:864–865, 2007.
16. Timothy R. Rebbeck, Mara Elena Martnez, Thomas A. Sellers, Peter G. Shields, Christopher P. Wild, and John D. Potter. Genetic variation and cancer: Improving the environment for publication of association studies. *Cancer Epidemiology Biomarkers & Prevention*, 13:1985–1986, 2004.
17. Joke Reumers, Joost Schymkowitz, Jesper Ferkinghoff-Borg2, Francois Stricher1, Luis Serrano1, and Frederic Rousseau. Snpeffect: a database mapping molecular

phenotypic effects of human non-synonymous coding snps. *Nucleic Acids Res.*, 33:D527–D532, 2005.

18. Yvan Saeys, Inaki Inza, and Pedro Larranaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, 2007.
19. Paul Shannon, Andrew Markiel, Owen Ozier, Nitin S. Baliga, Jonathan T. Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker. Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome Res.*, 13:2498–2504, 2003.
20. D. Shriner, L.K.Vaughan, M.A. Padilla, and H.K. Tiwari. Problems with genome-wide association studies. *Science*, 316:1840–1842, 2007.
21. Ola Spjuth, Tobias Helmus, Egon L Willighagen, Stefan Kuhn, Martin Eklund, Johannes Wagener, Peter Murray-Rust, Christoph Steinbeck, and Jarl ES Wikberg. Bioclipse: an open source workbench for chemo- and bioinformatics. *BMC Bioinformatics*, 8(59), 2007.
22. Ingenuity Systems. Ingenuity pathways analysis, 2007.
23. C. Szalai. Genomic investigation of asthma in human and animal models. In A. Falus, editor, *Immunogenomics and Human Disease*, pages 419–441. Wiley, London, 2005.
24. Alice S. Whittemore. Genetic association studies: Time for a new paradigm? *Cancer Epidemiology Biomarkers & Prevention*, 14:1359, 2005.
25. D.J. Wilkinson. Bayesian methods in bioinformatics and computational systems biology. *Briefings in Bioinformatics*, 8(2):109–116, 2007.
26. Y. Zhang and J. S. Liu. Bayesian inference of epistatic interactions in case-control studies. *Nature genetics*, 39(9):1167–1173, 2007.

BioGateway: enabling Semantic Systems Biology

Erick Antezana[1,2], Ward Blonde[1,2], Mikel Egaña[3], Robert Stevens[3], Bernard De Baets[4] Vladimir Mironov[5], and Martin Kuiper[1,2,5]

[1] Dept. of Plant Systems Biology, VIB, Gent, Belgium
[2] Dept. of Molecular Genetics, Ghent University, Gent, Belgium
[3] School of Computer Science, University of Manchester, UK
[4] Dept. of Mathematics, Biometrics and Process Control, Ghent University, Belgium
[5] Dept. of Biology, Norwegian University of Science and Technology, Norway
{erant|wablo|vlmir|makui}@psb.ugent.be
{eganaarm|stevensr}@cs.man.ac.uk
bdebaets@ugent.be

Abstract. The growing number of bio-ontologies begs for tools to interact with them. Such tools are essential to exploit what ontologies can offer in terms of standardization of terminologies. In addition, ontologies can be used for data integration and hypotheses generation, key elements in a systems biology approach. We implemented an integrated resource, named BioGateway, comprising the entire set of the OBO foundry candidate ontologies, the whole set of GOA files, the SwissProt protein collection and several in-house ontologies. The BioGateway provides a single entry point to exploit these resources with queries through SPARQL and constitutes a step towards a semantic integration of biological data. Access to the system and supplementary information (a tutorial, a listing of the data sources in RDF, and sample queries) can be found at http://www.semantic-systems-biology.org/biogateway

1 Rationale

The biosciences need a versatile and comprehensive knowledge integration framework. Although the concept of a portal is useful and several portals are currently in use providing significant amounts of data and information, it is rather difficult to ask simple questions e.g. like: "*which human diabetes-related proteins are located in the nucleus (of a part of it) and interacting with proteins related to pancreatic cancer*". Such questions need integration of multiple orthogonal sources of information at a basic level. This integrated knowledge resource may even allow the deployment of advanced computational reasoning approaches to generate new hypotheses about the functionality of biological systems, enabling a new concept that we have named *Semantic Systems Biology.* Here, we introduce an integrated knowledge repository named BioGateway, which enables the exploration and a combined querying of the entire set of the candidate OBO foundry ontologies, the GOA files, the SwissProt repository as well as in-house ontologies. The BioGateway provides a single entry point for exploiting these resources and constitutes a step towards a semantic web integration of biological data.

Reaction Kernels for Metabolic Modelling*

Katja Astikainen, Esa Pitkänen, and Juho Rousu**

Department of Computer Science, University of Helsinki

Abstract. Enzymes are the workhorses of living cells, producing energy and building blocks for cell growth as well as participating in maintaining and regulation of the metabolic states of the cells. Reliable assignment of enzyme function, that is, the biochemical reactions catalyzed by the enzymes, is a prerequisite of high-quality metabolic reconstruction and the analysis of metabolic fluxes.

Existing prediction tools of enzyme function are tied to set of functions already described in biological databases, with no capabilities of predicting previously unknown enzymatic function. Hence it can be argued their use is limited in, e.g., metabolic reconstruction of new organisms that have no close relatives with function annotation. We propose tackling this deficiency via structured output prediction with reaction kernels, a setup that allows us to interpolate and extrapolate in the space of enzymatic function.

We discuss the properties that a successful reaction kernel should have and the similarity notions that arise from the properties. First, different kernels may be designed to measure the similarity of reactant molecules and the reaction mechanism. Second, sensitivity to reaction direction may be a useful or a harmful properties, depending on the application. Based on these notions, we devise a family of reaction kernels to be used in metabolic modelling tasks. All of the presented reaction kernels are very efficient to compute, given an underlying kernel for the reactant molecules.

We illustrate the potential of these kernels to capture different similarity notions, and to complement the standard EC taxonomy for reaction classification.

* This work has been supported Academy of Finland under the grant 110514 (UR-ENZYMES), in part by Centre of Excellence Algodan (grant: 118653), and by the ICT Programme of the European Community, under the PASCAL2 Network of Excellence.

** Corresponding author. email: juho.rousu@cs.helsinki.fi

Microarray design using a kernel unsupervised feature selection technique

Justin Bedo

NICTA, Melbourne VIC 3010, Australia,
justin.bedo@nicta.com.au

The problem studied herein is to design a microarray plate by choosing a subset of clones from a large initial pool. The array must remain as general as possible and not be tailored towards specific phenotypes. As such, this is an unsupervised selection problem.

An Unsupervised feature selection method By the Hilbert-Schmidt independence criterion (UBHSIC) – a dependence measure between two random variables that is closely related to kernel target alignment and maximum mean discrepancy – is proposed and evaluated for this task on three cancer genomics datasets: the Alon colon cancer dataset, the van 't Veer breast cancer dataset, and a multiclass cancer of unknown primary (CUP) dataset. The multiclass CUP dataset is an ideal dataset to study as the goal is to select a small subset of features for the development of a clinical test.

Results

The effects and performance of several kernels (the Radial Basis Function (RBF), Linear, Polynomial, and a variance kernel) were evaluated on the various datasets. The variance kernel was chosen to encode a preference for uncorrelated features. Each dataset was analysed by applying UBHSIC with the various kernels to reduce the full dataset followed by supervised classification.

In summary, the results show that unsupervised pre-filtering does not degrade the classification performance and can actually improve the performance. Aggressive feature reduction down to 50 features for the two-class datasets and 100 features for the CUP dataset showed surprisingly good performance, suggesting that the full datasets contains significant redundancy and can be highly compressed without significant loss of performance.

Conclusions

The UBHSIC method was evaluated on several bioinformatics datasets and demonstrated good performance; the classification performance after pre-filtering using UBHSIC was equivalent or better than the performance obtained using the full dataset.

The high level of classification performance observed after unsupervised selection strongly suggests shifting to a lower resolution platform by selecting a subset of clones using UBHSIC is a viable option.

Module extraction in autoregressive models : application to gene regulatory networks inference

Nicolas Brunel [1,2], Yousri Slaoui [1], Florence d'Alché-Buc [1,3]

1-IBISC CNRS fre 3190, Université d'Evry-Val d'Essonne and Genopole, France.
2-ENSIIE, Evry, France.
3-URA CNRS 2171, Institut Pasteur, Paris, France.
Emails : brunel,yslaoui,dalche@ibisc.fr

Complex regulatory mechanisms at work in the cell are assumed to involve different subsets of genes, mRNA and proteins that behave more or less independently, implementing different biological functions. Under this biological assumption, models of gene regulatory networks should incorporate the notion of subnetworks or modules. We propose two new algorithms of module extraction in first order autoregressive models estimated from data. The first one consists in two steps : thresholding the estimated transition matrix using cross-validation [1] and then searching for an appropriate permutation to get a block-diagonal matrix with Theis' algorithm [2]. The second one consists in three steps : thresholding the empirical variance $\mathbb{E}\left[X_{t-1}X_{t-1}^T\right]$ (where X_t is the state vector) and the empirical covariance $\mathbb{E}\left[X_tX_{t-1}^T\right]$, estimating the transition matrix from the product of the thresholded empirical covariance and the pseudo-inverse of the thresholded empirical variance and finally identically to the previous described algorithm, searching for an adequate permutation to get a block-diagonal matrix. We sucessfully tested the two methods on simulated data and on gene expression kinetics of human keratinocytes during the switch between proliferation and differentiation [3]. Comparison between these two methods and with related existing methods [4,5] show that the first approach outperforms significantly the others.

Reference

1. Bickel, P. J. and Levina, E. Covariance regularization by thresholding. Ann. Statist. To appear. 2008.
2. Theis, F.J. Towards a general independent subspace analysis. In Proc. NIPS 2006.
3. d'Alché-Buc, F., Ambroise, C., Frouin, V., M-A. Debily, Vrain, C. ANR Project report 2007, 2007.
4. Friedman, J, Hastie, T and Tibshirani, R. Sparse inverse covariance estimation with the graphical lasso. Biostatistics. 9(3) :432-441 2008.
5. Banerjee, O, El Ghaoui, L. E, d'Aspremont, A. Model Selection Through Sparse Maximum Likelihood Estimation for Multivariate Gaussian or Binary Data. Bulletin of American Mathematical Society. 9, 485-516 2008.

Acknowledgement and contribution This project has been funded by Genopole (ATIGE) and ANR (project ARA Genomic data to graph structure : G2GS). The two first authors equally contribute to this work.

Efficient Query-Driven and Global Biclustering of Gene Expression Data Using Probabilistic Relational Models

Tim Van den Bulcke[1], Hui Zhao[2], Kristof Engelen[2], Tom Michoel[34], Bart De Moor[1], Kathleen Marchal[2]

[1] Dept. of Electrical Engineering (ESAT-SCD), Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, 3001 Leuven, Belgium.
[2] Dept. Microbial and Molecular Systems (CMPG), Katholieke Universiteit Leuven, Kasteelpark Arenberg 20, 3001 Leuven, Belgium.
[3] Department of Plant Systems Biology, VIB, Technologiepark 927, B-9052 Gent, Belgium.
[4] Department of Molecular Genetics, UGent, Technologiepark 927, B-9052 Gent, Belgium.

Abstract. Biclustering is an increasingly popular technique to identify gene regulatory modules that are linked to biological processes. We describe a novel method, called *Pro*Bic, that was developed within the framework of Probabilistic Relational Models (PRMs). *Pro*Bic is an efficient biclustering algorithm that simultaneously identifies a set of potentially overlapping biclusters in a gene expression dataset. The algorithm can be applied in both a query-driven and a global setting.
The results on a wide range of synthetic datasets show that *Pro*Bic successfully identifies biclusters under various levels of noise, overlap and missing values and this in both the query-driven and global setting. Additional expert knowledge can be introduced through a number of prior distribution parameters. Our results on synthetic data show that PRMs can be used to identify overlapping biclusters in an efficient and robust manner.

Key words: biclustering, probabilistic relational model, gene expression, regulatory module, expectation-maximization.

The reconstruction of the cellular signaling pathways is one of the foremost challenges in systems biology [1]. While a large amount of high throughput *'omics* data are available, the reconstruction of this signaling network still remains a highly underdetermined problem. Currently, insufficient data is available to uniquely identify all the interactions and their parameters in the model. However, regulatory networks exhibit a modular organization [2], an aspect that is successfully exploited in a number of biclustering and module inference methods [3]. *Biclustering* algorithms for gene expression data [4] perform simultaneous clustering in both the gene and condition dimensions. The result is a subset of genes that are coexpressed under a subset of conditions.

The *Pro*Bic model uses a hybrid query-driven and model-based approach. This allows researchers to both identify biclusters using a global approach and to incorporate prior knowledge by performing directed queries around genes of interest. Although some other algorithms also allow for query-driven searches such as for instance, the iterative signature algorithm [5], Gene Expression Mining Server [6], Gene Recommender [7] and QDB [8], they do not combine the advantages of the query-driven search with a model based approach for identifying overlapping biclusters. We present an alternative approach to identify overlapping biclusters in gene expression data using Probabilistic Relational Models [9–11]. *Pro*Bic can be applied in both a query-driven and a global setting. Moreover, *Pro*Bic was designed such that it is easily extensible towards additional data types.

An extensive evaluation of the algorithm was performed on synthetic data to investigate the behavior of the algorithm under various parameter settings and input data. Firstly, we tested the robustness of the algorithm w.r.t. noise and amount of missing values. Results for synthetic datasets with 500 genes and 200 conditions show that perfect bicluster reconstruction was achieved for bicluster noise levels up to 70% of the background noise. Secondly, the presence of up to 60% of missing values in the dataset does not interfere with the detection of the true bicluster genes and conditions (a precision and recall of about 100% is obtained for both the genes and conditions) for lower bicluster noise levels (< 0.5). Even in the presence of high noise levels (1.0), the presence of up to 20% missing values does not considerably deteriorate the algorithms recall and precision (recall levels of 0.9 and 0.88 for the genes and conditions respectively). We show that prior knowledge in the form of a set of query genes guides the algorithm towards biclusters of interest to a biologist. Moreover, the bicluster identification using query genes is quite robust as the set of query genes can contain several 'noisy' genes that are not part of the bicluster of interest, a situation that often occurs in practice.

In conclusion, *Pro*Bic is an efficient biclustering algorithm that simultaneously identifies a set of overlapping biclusters in a gene expression dataset. It can be used in both a query-based and a global mode. Experiments on synthetic data showed that biclusters are successfully identified under various settings, both in the query-driven and the global setting.

References

1. Greenbaum, D., Luscombe, N.M., Jansen, R., Qian, J., Gerstein, M.: Interrelating different types of genomic data, from proteome to secretome: 'oming in on function. Genome research **11**(9) (September 2001) 1463–1468
2. Tanay, A., Sharan, R., Kupiec, M., Shamir, R.: Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data. Proceedings of the National Academy of Sciences U.S.A **101**(9) (March 2004) 2981–2986
3. Van den Bulcke, T., Lemmens, K., Van de Peer, Y., Marchal, K.: Inferring transcriptional networks by mining 'omics' data. Current bioinformatics **1** (2006)
4. Cheng, Y., Church, G.M.: Biclustering of expression data. Proc Int Conf Intell Syst Mol Biol **8** (2000) 93–103
5. Bergmann, S., Ihmels, J., Barkai, N.: Iterative signature algorithm for the analysis of large-scale gene expression data. Physical review. E, Statistical, nonlinear, and soft matter physics **67**(3 Pt 1) (March 2003)
6. Wu, C.J., Kasif, S.: Gems: a web server for biclustering analysis of expression data. Nucleic Acids Res **33**(Web Server issue) (July 2005)
7. Owen, A.B., Stuart, J., Mach, K., Villeneuve, A.M., Kim, S.: A gene recommender algorithm to identify coexpressed genes in c. elegans. Genome Res. **13**(8) (August 2003) 1828–1837
8. Dhollander, T., Sheng, Q., Lemmens, K., De Moor, B., Marchal, K., Moreau, Y.: Query-driven module discovery in microarray data. Bioinformatics **23**(19) (October 2007) 2573–2580
9. Koller, D., Pfeffer, A.: Probabilistic frame-based systems. Proc. AAAI, Madison (July 1998) 580–587
10. Friedman, N., Getoor, L., Koller, D., Pfeffer, A.: Learning probabilistic relational models. (1999) 1300–1309
11. Getoor, L., Friedman, N., Koller, D., Taskar, B.: Learning probabilistic models of relational structure, Morgan Kaufmann, San Francisco, CA (2001) 170–177

A kernel-based method for the integration of high-throughput data sources to improve clinical decision support in cancer

Anneleen Daemen, Olivier Gevaert and Bart De Moor

Department of Electrical Engineering (ESAT)
Katholieke Universiteit Leuven, Leuven, Belgium

Motivation: Although microarray technology allows the investigation of the complete transcriptomic make-up of a tumour in one experiment, the transcriptome doesn't tell the full story due to alternative splicing or post-translational modifications as well as the influence of cell type and pathological conditions (e.g. cancer) on translation. The likeliness that multiple views contain different, partly independent and complementary information makes the fusion of various types of genome-wide data an increasingly important topic in bioinformatics. The current increase in the amount of available data emphasizes the need for a methodological framework to integrate different omics data sources.

Method: Kernel methods are increasingly used in bioinformatics due to their reliability, accuracy and computational efficiency. We propose a kernel-based framework for the development of classifiers in clinical decision support in which high-throughput data sources can be combined over time and multiple levels in the genome. For the fusion of multiple data sets, an intermediate integration approach was opted for in which each data set is represented by a kernel matrix before training a classifier on the explicitly heterogeneous kernel matrix. Because in two-class problems data sets are often skewed such that the contribution of false negative and false positive errors are not balanced, we used as supervised classification algorithm the weighted Least Squares Support Vector Machine (wLS-SVM), an extension of the standard SVM which takes the unbalancedness of data sets into account. Selection of the most relevant features was embedded in training the wLS-SVM models.

Results: This framework has been applied on microarray and proteomics data, gathered at two timepoints during preoperative treatment of patients with rectal cancer. Different grades of integration (over time and over multiple levels in the genome) were considered and compared to models built on individual data sets. Two prognostic factors determined at moment of surgery could be predicted optimally with an accuracy of 94.4% using both microarray and proteomics data one week after start of therapy. Also for the two regression grades registered during surgery, models integrating data from multiple levels in the genome gave the best results (83.3% and 88.9%). The advance of the classification performance when considering diverse experimental data confirms the need for an integration framework presented on the poster.

Comparative analysis of gene expression across species

Ana Carolina Fierro[1], Peyman Zarrineh[1], Kristof Engelen[1], Mieke Verstuyf[2], Guy Eelen[2], Lieve Verlinden[3], and Kathleen Marchal[1]

[1]Department of Microbial and Molecular system, Katholieke Universiteit Leuven, Kasteelpark Arenberg 20, 3001 Leuven, Belgium
[2] Experimentele Geneeskunde en Endocrinologie, Katholieke Universiteit Leuven, 3000 Leuven, Belgium
[3] Laboratorium voor Experimentele Geneeskunde en Endocrinologie, Katholieke Universiteit Leuven, 3000 Leuven, Belgium

Gene expression analyses on model organisms have been widely used to study the genetic response to a certain stimulus, but this response can vary between species. A comparative study of gene expression levels between two species can give us valuable information about how conserved the mechanism is that responds to a given stimulus.

In this study we compared the response to vitaminD in human and mouse based on the gene coexpression derived from microarray experiments. In order to distinguish between general differences due to the used species and cell type from those induced by the stimulus of interest (vitaminD), we performed for each organism a control and a treatment response experiment. We used the Differential Clustering Algorithm [1] to evaluate the degree of conservation of gene expression between both, human and mouse.

References

[1] Ihmels J, Bergmann S, Berman J, Barkai N. Comparative gene expression analysis by differential clustering approach: application to the Candida albicans transcription program. *PLoS Genet.* 2005 Sep;1(3):e39.

Model-based SNP set selection in study design using a multilevel, sequential, Bayesian analysis of earlier data sets

G. Hajós, P. Antal, Y. Moreau, Cs. Szalai, and A. Falus

Dept. of Measurement and Information Systems
Budapest University of Technology and Economics
Dept. of Electrical Engineering
Katholieke Universiteit Leuven
Inflammation Biology and Immunogenomics Research Group
Hungarian Academy of Sciences
Dept. of Genetics, Cell- and Immunobiology
Semmelweis University, Hungary

Partial genome screening studies (PGSS) and genome-wide association studies (GWAS) are essential tools in genetic association studies (GAS). Their role are complementary because of their cost w.r.t. sample size and dimensionality, which will probably remain in the near future. In the design of a fixed cost PGSS for a given biomedical problem, central questions are the selection of SNPs, plates, and the question of sample sizes. Interestingly, the selection of a SNP set based on prior knowledge can be an equally important issue in the data analysis of GWA studies to diminish the problem of multiple testing.

In SNP selection there are many aspects to consider: the technological constraints of the high-throughput device (GC content, primer design, etc.), the haplotype structure of the target population (e.g., see [6, 10, 8]), the domain knowledge about genes (e.g., see [16, 18, 1]), and the possible functional effect of SNPs (e.g., see [12, 5, 14, 15, 11, 7]). To support these aspects, especially to take into consideration the measurement device at the Department of Genetics, Cell- and Immunobiology of Semmelweis University, we implemented a decision support system for filtering, univariate scoring, and multivariate scoring of SNPs. Currently the multivariate property reflects only the haplotype dependency and joint coverage of target regions of the SNPs.

An important open problem is the principled use of earlier data sets in a subsequent study design. In the poster we present the use of sequential Bayesian data analysis of earlier data sets in study design (for the Bayesian approach in GAS, see e.g. [20, 4, 17, 22, 21, 13]). Specifically, we illustrate the use of the results of Bayesian Multilevel Analysis (BMLA) for sample size selection and SNP selection. The BMLA uses Bayesian networks [9] for the analysis of relevance at the pairwise, multivariate, and multivariate-interactionist levels [2, 3].

Finally, we discuss the use of the Bayesian approach to perform a value of further information analysis in designing PGSSs by predicting the effects of a future data set with size N. The genomics of asthma will serve as an application domain [19].

References

1. Bonis J andFurlong LI and Sanz F. Osiris: a tool for retrieving literature about sequence variants. *Bioinformatics*, 22(20):2567–9, 2006.
2. P. Antal, A. Millinghoffer, G. Hullám, G.Hajós, Cs. Szalai, and A. Falus. A bioinformatic platform for a Bayesian, multiphased, multilevel analysis in immunogenomics. In D.R.Flower M.N.Davies, S.Ranganathan, editor, *Bioinformatics for Immunomics*, pages –. Springer, 2008.
3. P. Antal, A. Millinghoffer, G. Hullám, Cs. Szalai, and A. Falus. A bayesian view of challenges in feature selection: Feature aggregation, multiple targets, redundancy and interaction. *JMLR Proceeding*, 42:–, 2008.
4. D. J. Balding. A tutorial on statistical methods for population association studies. *Nature*, 7:781–91, 2006.
5. Worth CL, Bickerton GR, Schreyer A, Forman JR, Cheng TM, Lee S, Gong S, Burke DF, and Blundell TL. A structural bioinformatics approach to the analysis of nonsynonymous single nucleotide polymorphisms (nssnps) and their relation to disease. *J Bioinform Comput Biol.*, 5(6):1297–318, 2007.
6. The International HapMap Consortium. A second generation human haplotype map of over 3.1 million snps. *Nature*, 449:851–862, 2007.
7. Richard J Dobson, Patricia B Munroe, Mark J Caulfield, and Mansoor AS Saqi. Predicting deleterious nssnps: an analysis of sequence and structural attributes. *BMC Bioinformatics*, 7:217, 2006.
8. Hong Xu et al. Snpselector: a web tool for selecting snps for genetic association studies. *Bioinformatics*, 21(22):263–5, 2005.
9. N. Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, 303(5659):799 – 805, 2004.
10. Barrett JC, Fry B, Maller J, and Daly MJ. Haploview: analysis and visualization of ld and haplotype maps. *Bioinformatics*, 21(2):263–5, 2005.
11. Byoung-Chul Kim, Woo-Yeon Kim, Daeui Park, Won-Hyong Chung, Kwang sik Shin, and Jong Bhak. Snp@promoter: a database of human snps (single nucleotide polymorphisms) within the putative promoter regions. *BMC Bioinformatics*, 9, 2008.
12. Bao L, Zhou M, and Cui Y. nssnpanalyzer: identifying disease-associated nonsynonymous single nucleotide polymorphisms. *Nucleic Acids Res.*, 33:W480–2, 2005.
13. Harold P. Lehmann and Steven N. Goodman. Bayesian communication, a clinically significant paradigm for electronic publication. *Journal of the American Medical Informatics Association*, 7:254–266, 2000.
14. Wang P, Dai M, Xuan W, McEachin RC, Jackson AU, Scott LJ, Athey B, Watson SJ, and Meng F. Snp function portal: a web database for exploring the function implication of snp alleles. *Bioinformatics*, 22(14):e523–9, 2006.
15. Joke Reumers, Joost Schymkowitz, Jesper Ferkinghoff-Borg2, Francois Stricher1, Luis Serrano1, and Frederic Rousseau. Snpeffect: a database mapping molecular phenotypic effects of human non-synonymous coding snps. *Nucleic Acids Res.*, 33:D527–D532, 2005.
16. Aerts S, Lambrechts D, Maity S, Van Loo P, Coessens B, De Smet F, Tranchevent LC, De Moor B, Marynen P, Hassan B, Carmeliet P, and Moreau Y. Gene prioritization through genomic data fusion. *Nature Biotechnol.*, 24(5):537–44, 2006.
17. Yvan Saeys, Inaki Inza, and Pedro Larranaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, 2007.
18. Ingenuity Systems. Ingenuity pathways analysis, 2007.

19. C. Szalai. Genomic investigation of asthma in human and animal models. In A. Falus, editor, *Immunogenomics and Human Disease*, pages 419–441. Wiley, London, 2005.
20. Alice S. Whittemore. Genetic association studies: Time for a new paradigm? *Cancer Epidemiology Biomarkers & Prevention*, 14:1359, 2005.
21. D.J. Wilkinson. Bayesian methods in bioinformatics and computational systems biology. *Briefings in Bioinformatics*, 8(2):109–116, 2007.
22. Y. Zhang and J. S. Liu. Bayesian inference of epistatic interactions in case-control studies. *Nature genetics*, 39(9):1167–1173, 2007.

More is Better: Ranking with Multiple Targets for Biomarker Discovery

Dragi Kocev, Ivica Slavko, and Saso Dzeroski

Jozef Stefan Institute, Slovenia
{Dragi.Kocev,Ivica.Slavkov,saso.dzeroski}@ijs.si

The process of biomarker discovery is equivalent to the process of feature ranking and selection in machine learning. Each marker has a relevance measure assigned to it by a ranking algorithm. Typically, the ranking is produced with respect to a single target variable (e.g. outcome of a disease). But, the clinical data for a patient is much more complex and it contains multiple variables of interest.

Here, we address the problem of feature ranking in the context of multiple targets. In particular, we propose an extension of feature ranking with Random Forests (RFs) by enabling the method to handle multiple target variables simultaneously. The feature importance measure is calculated by randomly permuting the values of the features and measuring the out-of-bag (OOB) error estimates. The rationale is that if a feature is important for the target concepts it should have an increased error rate when its values are randomly permuted.

We apply the proposed method for feature ranking with multiple targets to Neuroblastoma microarray data associated with clinical data containing multiple variables of interest. We produce ranked lists of genes with respect to different (single) clinical parameters and compare these ranked lists with the one produced by considering multiple target variables simultaneously. We compare the ranked lists by using so-called average testing error curves (ATEs), which give us an estimate of the predictive performance of the highly ranked genes (markers). The results show an increase in the predictive performance of the highly ranked genes, when considering multiple target variables as compared to the ones from the ranked lists for each target variable individually. It is important to note that the same set of markers produced by the multiple target approach can be used for predicting the different clinical variables instead of having a different set of markers for each one.

In summary, we consider the process of biomarker discovery from a perspective of single vs. multiple target variables. The intuition behind using multiple target variables simultaneously comes from the usual complexity of the diseases under consideration (e.g. cancer) and the associated multi-variable patient clinical data. Our initial results show that the multiple target approach is beneficial as compared to the single target variable approach. The produced ranked list of biomarkers is more accurate, in terms of predictive performance, and it can be applied to each of the target variables separately.

Strategies for computational transcription factor binding site discovery in humans

Geoff Macintyre[1], James Bailey[1], Adam Kowalczyk[1] and Izhak Haviv[2]

[1]National ICT Australia and The University of Melbourne,
Parkville, Victoria 3010, Australia
[2]Baker IDI Heart and Diabetes Institute ,
250 Kooyong Road Caulfield, Victoria 3162, Australia

Computational approaches for *de novo* discovery of Transcription Factor Binding Sites (TFBSs) are commonly focused around the Transcription Start Site (TSS), typically 1kb to 10kb upstream. This promoter proximal search space is largely due to the study of model organisms such as yeast, where the majority of binding sites lie close to the TSS. This search space may not be suitable in humans.In fact, a study into a cis-acting regulator causing preaxial polydactyly demonstrated a case where a TF was empirically determined to act over a distance of around 800kb [1].

Recently, chromatin immunoprecipitation coupled with high-throughput sequencing technologies has provided the ability to empirically observe the behavior of particular Transcription Factors (TFs) and map their binding sites genome-wide, providing a valuable tool for the validation of TFBS discovery methods and elucidation of gene regulatory networks.

We compiled a study of four different ChIP-PET datasets identifying TFBSs for ER [2], STAT1 [3], Myc [4] and p53 [5]. We matched TFBSs with their target genes, highlighting typical genomic distances of TF action. We observed the following:
(i) An average of only 27% of genes had a TFBS within 10000bp of their TSSs
(ii) All TFs showed equal distributions of upstream vs downstream binding
(iii) An average of 33% of genes had TFBSs beyond 200000bp of their TSSs
(iv) 55% of the binding site target gene pairs had genes residing between the binding site and the TSS of the target gene

Conclusion. Computational identification of TFBSs must shift focus to genomic regions outside the proximity of the TSS. Careful consideration also needs to be given to genes deemed as direct targets of a TF. Unfortunately, with large input sequences, the majority of current computational TFBS discovery approaches suffer from many false positive predictions. With intelligent use of the increasing amount of high-throughput genomic data, we believe that current approaches can be improved to scale well when applied to the large genomic search spaces required in humans.

References

1. Lettice LA, *et al.*, PNAS 2002, 99:75487553.
2. Lin CY, *et al.*,PLoS Genetics 2007, 3:867885.
3. Robertson G, *et al.*, Nat Meth 2007,4:651657.
4. Zeller KI, *et al.*, PNAS 2006, 103:1783417839.
5. Wei CL, *et al.*, Cell 2006, 124:207219.

Gene Ontology assisted microarray clustering and its application to cancer

Geoff Macintyre[1,2], James Bailey[1,2], Daniel Gustafsson[4], Alex Boussioutas[3],
Izhak Haviv[1,5], and Adam Kowalczyk[2]

[1] University of Melbourne, Victoria, Australia
[2] National ICT Australia, Victorian Research Lab, Australia
[3] Peter MacCallum Cancer Centre, East Melbourne, Victoria, Australia
[4] La Trobe University, Victoria, Australia
[5] BakerIDI, Melbourne, Australia

Clustering gene expression profiles can facilitate the identification of the biological program causing genes co-expression. Standard clustering methods may fail to capture weak expression correlations causing genes in the same biological process to be grouped separately. To improve the clustering process additional information such as The Gene Ontology (GO)[1] can be introduced to ensure genes with similar function can be clustered together.

Previous attempts[2-5] at using the GO to cluster microarrays use similarity metrics between pairs of genes based on the GO graph structure. This approach suffers from two fundamental drawbacks: firstly, the abstraction of terms across each level of the ontology can be such that two genes with a single shared parent term, may be extremely diverse in terms of their specific function; secondly, having genes annotated to the same term does not necessarily imply they have similar function or share a biological pathway, in the context of their expression patterns.

To account for this we have developed GOMAC: Gene Ontology assisted MicroArray Clustering. GOMAC is a modified k-means clustering algorithm which incorporates GO information only when it is relevant to the gene's context, thus avoiding problems with irrelevant gene similarities. The key difference between our approach and previous attempts at clustering using the GO is that only terms that are statistically over-represented within a cluster are used to calculate the similarity between genes. This ensures that only GO terms within the genes expression context are used.

GOMAC was tested on a multi-class cancer dataset with respect to its usefullness in biological hypothesis generation. Our results demonstrate that incorporation of additional biological information into the microarray clustering process in a biologically justified manner, can enhance the interpretability of microarray data. Specifically, we show the potential of such a method to unravel the complex nature of the biological processes involved in cancer.

References

1. Ashburner, M., *et al.*, Nat. Genet. 2000, 25:2529.
2. Cheng, J., *et al.*,J. Bio-pharm. Stat. 2004, 14:687700.
3. Huang, D., *et al.*, Bioinf. 2006, 22:12591268.
4. Pan, W, *et al.*, Bioinf. 2006, 22:795801.
5. Boratyn, G.M, *et al.*, Bioinformation 2007, 1.

Ab initio gene prioritization

Daniela Nitsch and Yves Moreau

KU Leuven, ESAT-SCD, Belgium
{Daniela.Nitsch,yves.moreau}@esat.kuleuven.be

Genetic studies and high-throughput genome wide screens can identify genes and proteins that are candidate members for a biological process of interest (such as disease and pathways). The disadvantage of the screens is their identification of tens or hundreds of candidate genes. Aerts et al. (2006) developed gene prioritization methods that rank candidate genes based on their similarity to genes already associated with a disease or process, using multiple data sources (such as sequence, expression, literature, etc). However, this method cannot prioritize candidates if no similar genes can be identified a priori. This prevents the method from tackling truly innovative discoveries, when little is known about a disease. Currently, there are no well-established gene prioritization strategies without a set of training genes. The new strategy we are working on, consists of checking not only the expression of a candidate, but also of its ”partner” genes in a gene network derived from multiple sources. A strong candidate should have many partners that are differentially expressed, meaning that it belongs to a disrupted expression module.

A mouse model network was built using protein-protein interactions extracted from the STRING database (http://string.embl.de), a comprehensive dataset containing functional linkages (labeled with scores), for which the similarities between the single genes could be computed. Candidate genes with genes in their neighborhood having highly differentially expressed indices are strong candidates. Different kernel matrices were used as global distance measures to capture global relationship within the network, e.g. the Exponential diffusion Kernel (Konder et al. 2002), the Commute time Kernel (Fouss et al. 2006), the Von Neumann Kernel (Kandola et al. 2003). To improve the enormous computing time, the kernel matrices were approximated by the Incomplete Cholesky decomposition (Fine and Scheinberg 2001) and the Reduced Eigenvalue decomposition. The outcoming kernel matrix contains global relationships within the fully connected protein-protein interaction network. From this matrix an appropriate neighborhood for each candidate gene can be defined, i.e. which genes act as neighbors in the network.

Knowing the similarities between single genes in the network, the differentially expression index of the genes themselves, as well as the differentially expression index of adjacent genes can be considered, based on microarray experiments. We chose an expression dataset published in Battle et al. (2006) of a HNF4 knockout experiment, and added the expression data to the network. The candidate genes can then be ranked considering the similarity to highly differentially expressed neighbor genes.

In the future this method will be applied to congenital heart defects studies ongoing at Leuven University Hospitals.

Hidden Markov Models support array-based prediction of DNA copy number variants in Arabidopsis ecotypes

Michael Seifert[1], Ali Banaei[1], Jens Keilwagen[1], François Roudier [2], Vincent Colot[2],
Florian Michael Mette[1], Andreas Houben[1], Ivo Grosse[3], and Marc Strickert[1]

[1]Leibniz Institute of Plant Genetics and Crop Plant Research, Gatersleben, Germany
[2]École Normale Supérieure, Paris, France
[3]Martin Luther University, Institute of Computer Science, Halle, Germany

Contact: seifert@ipk-gatersleben.de

Natural variation of Arabidopsis thaliana ecotypes from Europe, Africa, and Africa is reflected to a substantial degree in their genome sequences. In this study, array comparative genomic hybridization (Array-CGH) is used to quantify the natural variation of different ecotypes at the DNA level. A new approach based on Hidden Markov Models (HMMs) is presented to predict copy number variants of such Array-CGH experiments. The HMM approach provides an improved genome-wide characterization of DNA segments with decreased or increased copy numbers in comparison to the routinely used SegMNT algorithm [1]. The identification of significantly altered chromosome fragments helps to establish a faithful genome-wide map of DNA copy number variants of different ecotypes. DNA copy number variants are further investigated by making use of mappings to the TAIR8 genome annotation. Particular enrichment of TAIR8 categories is assessed by resampling statistics. Another major benefit is that the presented method can be transferred to other array analyses, such as ChIP/chip studies [2].

[1] Roch NimbleGen, Inc. (2008). A Performance Comparison of Two CGH Segmentation Analysis Algorithms: DNACopy and segMNT. (http://www.nimblegen.com)

[2] Michael Seifert, Jens Keilwagen, Marc Strickert, and Ivo Grosse (2008): Utilizing promoter pair orientations for HMM-based analysis of ChIP-chip data, Lecture Notes in Informatics, GCB 2008, Dresden.

Progressive Clustering by Integration of Heterogenous Data From Multiple Sources for Target Gene Identification

Zerrin Sökmen[1], Volkan Atalay[1], and Rengül Çetin-Atalay[2]

[1] Department of Computer Engineering, Middle East Technical University, Ankara TURKEY
zerrin.sokmen,volkan@ceng.metu.edu.tr
[2] Department of Molecular Biology and Genetics, Bilkent University, Ankara TURKEY
rengul@bilkent.edu.tr

Abstract. Traditional approaches to analyze microarray data are useful for describing changes in gene expression, however they are of limited use to describe cellular responses in the context of available biological knowledge. In order to find out gene sets that are biologically more meaningful, biological information should be integrated during the analysis of microarray data or after the identification of differentially expressed gene lists. Therefore recent effort focuses on the discovery of biological pathways rather than individual gene analysis. Several gene prioritization methods attempt to determine the similarity between candidate genes and genes known to play a role in defined biological processes or diseases [1, 2]. In this study, the aim was to extract the clusters of functionally related genes over the short-time series microarray data based on unsupervised methods and to automatically perform biological annotation of the extracted clusters. We used both public annotations and function predictions coming from a function prediction tool (SPMap) and then clustered differentially expressed gene lists by applying a progressive clustering method [3]. Original microarray data was composed of 54000 probe sets of 3 days expression samples for 2 experimental conditions, HepG2 and HepG2-2.2.15, and experimented in our laboratory. The aim of microarray experiment is to observe selenium deficiency in hepatocellular carcinoma cells under oxidative stress. Original data was evaluated as short-time series data due to limited number of observations and their time dependence. Initially, short-time series microarray expression data was clustered according to similar expression profiles by applying k-means algorithm with $k = 100$ setting. Gene expressions were represented with piece-wise linear functions and the difference between the slopes of these functions was used as distance measure in k-means clustering phase. After completion of k-means clustering, we selected 12 clusters as candidate patterns since their expression profiles were consistent with pre-defined expression profiles which were supposed to be responsible of resistance to oxidative stress. In the second phase, we integrated Gene Ontology (GO) annotations and expression data of the genes in these clusters to obtain more biological information related with these genes. We preferred to use an "information content" based distance measure to represent the semantic distance of two genes in the GO hierarchy [4]. SPMap was applied to predict GO annotations of genes with unknown functions. The genes with known protein sequences were given to SPMap which predicts GO terms of a given protein sequence by considering 300 "molecular function" terms of GO hierarchy. The semantic and expression distances of genes were combined using a weighted scheme. The combined distance matrix contains only pairwise distances between genes. Therefore instead of using traditional clustering algorithms we applied a graph partitioning method: spectral clustering algorithm ($k = 6$ setting). After completion of spectral clustering, each resulting sub-component contained 3-10 genes which share similar patterns in terms of both gene expression level and molecular function.

References

1. Aerts, S., Lambrechts, D., Maity, S., Loo, P.V., Coessens, B., Smet, F.D., Tranchevent, L.C., Moor, B.D., Marynen, P., Hassan, B., Carmeliet, P., Moreau, Y.: Gene prioritization through genomic data fusion. Nat.Biotech. **24** (2006) 537–544
2. Bie, T.D., Tranchevent, L.C., van Oeffelen, L.M.M., Moreau, Y.: Kernel based data fusion for gene prioritization. Bioinformatics **23** (2007) i125–i132
3. Sarac, O.S., Yuzugullu, O., Atalay, R., Atalay, V.: Subsequence based feature map for protein function classification. J. Comp. Bio. and Chem. **32**(2) (2008) 122–130
4. Resnik, P.: Semantic similarity in a taxonomy: an information-based measure and its application to problems of ambiguity in natural language. J. Artif. Intel. Res. **11** (1999) 95–130

A data mining platform for systems biology and biomarker discovery

Olivier Stern[1], Raphaël Marée[2], Christophe Van Huffel[3], Jean-Francois Laes[4], Carine Michiels[5], Lionel Flamant[5], Véronique Mainfroid[6], Daisy Flamez[7], Louis Wehenkel[1], and Pierre Geurts[1]

[1] Department of Electrical Engineering and Computer Science, GIGA-Research,
Université de Liège, Belgium
{Olivier.Stern,P.Geurts}@ulg.ac.be
[2] GIGA Management, Bioinformatics Platform, Université de Liège, Belgium
[3] KeyMarker project - BioWin, Namur, Belgium
keymarker@biowin.org
[4] DNAvision SA, Gosselies, Belgium
[5] Unité de Recherche en Biologie Cellulaire, Facultés Universitaires Notre-Dame de la Paix, Namur, Belgium
[6] Eppendorf Array Technologies SA, Namur, Belgium
[7] Laboratory of Experimental Medicine, Université Libre de Bruxelles, Belgium

The general goal of this research is to develop a bioinformatic strategy to discover new candidate biomarkers for some specific disease, by integrating biological knowledge available in public databases and experimental data related to this disease obtained from high-throughput instrumentations, such as transcriptomic (microarray), proteomic (mass spectrometry), and genomic (SNP) data. The resulting strategy will be implemented in a generic and flexible software platform that will allow biologists to easily instantiate this strategy on their own datasets. This research is part of the Keymarker project (Biowin), whose general goal is to identify biomarkers for molecular imagery.

Currently, the developed software is decomposed into several independent modules. A filtering module allows to (manually) select genes/proteins using simple rules based on biological annotations and/or experimental data. A clustering module automatically identifies groups of genes/proteins that behave similarly in one or several experiments. A classification module exploits supervised classification methods to select groups of genes/proteins whose behaviour allows to discriminate between several pre-defined biological conditions. Finally, an enrichment analysis module helps to better characterize the groups of relevant biomarkers highlighted by the other modules (by exploiting existing annotations of the corresponding genes or proteins in biological databases).

One of the originality of the software is that it integrates all steps in a common platform, making thus easy the interaction between the different modules. The platform is accessible as a web service as well as a standalone application. Future modules will be added to allow the joint analysis of different types of expression data (mRNA, microRNA, transcription factors...) and the integration of different experimental data sources (transcriptomic, proteomic, and genomic).

Acknowledgements This work is supported by ”Pôles de compétitivité de la région Wallonne” as part of the KeyMarker - BioWin project.

Using Bigram Language Model for Protein Name Recognition

Serhan TATAR, Ilyas Cicekli

Department of Computer Engineering, Bilkent University 06800 Bilkent, Ankara, Turkey
{statar, ilyas}@cs.bilkent.edu.tr

1 Motivation

As one of the basic tasks in automatic discovery and extraction of information from biological texts, protein name extraction is still a challenge. Extracting protein names from unstructured texts is a prerequisite for the increasing demand in automatic discovery and extraction of information from biological texts. Locating the information on different levels can be seen as a layered structure and this layered structure makes different extraction tasks interdependent. Because the output of a task at a layer is input to the next layer, the success of a former task affects the performance of the others. For instance, how well we locate the protein names in a text has an impact on how well we find the interactions between the proteins.

2 Method

In order to identify protein names, we study using bigram language model, a special case of N-gram which is used in various areas of statistical natural language processing, along with the hierarchically categorized syntactic word types. We determine 21 syntactic token types categorized under five main classes to generalize protein names: *single*, *abbreviation*, *delimiter*, *regular*, and *other*.

After learning the necessary model parameters, a probability estimate is produced for every possible fragment in the test data. We use sliding-window technique to determine the fragments. Fragments with the highest likelihood, exceeding a certain threshold value, are extracted as protein names.

3 Results

Table 1 compares performance values of our method (Bigram) with the values published for several methods. Our method has a comparable performance to the others with respect to F-score. The comparison also shows that our method is effective and competitive.

Table 1. Comparison of methods for protein name extraction.

	Recall	**Precision**	**F-score**
Bigram	67.5%	60.2%	63.6 %
YAPEX [1]	59.9%	62.0%	61.0 %
SemiCRF [2]	76.1%	58.9%	66.1 %
DictHMM [2]	45.1%	69.7%	54.8 %
Prob [3]	60.1%	66.9%	63.3 %

References

1. Proteinhalt i text, http://www.sics.se/humle/projekt/prothalt/
2. Kou, Z., Cohen, W. W., and Murphy, R. F. 2005. High-recall protein entity recognition using a dictionary. Bioinformatics 21, 1 (Jan. 2005), 266-273.
3. Seki, K. and Mostafa, J. 2003. A Probabilistic Model for Identifying Protein Names and their Name Boundaries. In Proceedings of the IEEE Computer Society Conference on Bioinformatics (August 11 - 14, 2003). CSB. IEEE Computer Society, Washington, DC, 251.

Gene prioritization through genomic data fusion

Léon-Charles Tranchevent[1], Stein Aerts[2,3], Bernard Thienpont[4], Peter Van Loo[1,2,5], Shi Yu[1], Bert Coessens[1], Roland Barriot[1], Steven Van Vooren[1], Bassem Hassam[2,3] and Yves Moreau[1]

[1] Department of Electrical Engineering ESAT-SCD,
Katholieke Universiteit Leuven (Belgium)
[2] Department of Human Genetics,
Katholieke Universiteit School of Medicine, Leuven (Belgium)
[3] Laboratory of Neurogenetics, Department of Molecular and Developmental Genetics, VIB, Leuven (Belgium)
[4] Center for Human Genetics, Katholieke Universiteit Leuven (Belgium)
[5] Human Genome Laboratory, Department of Molecular and Developmental Genetics, VIB Leuven (Belgium)

Abstract. Genome-wide experimental methods to identify disease genes (such as linkage analysis and association studies) often generate large list of candidate genes from which only a few are interesting. Endeavour (http://www.esat.kuleuven.be/endeavour), a web resource for the prioritization of genes, indicates which genes are the most promising ones. Our approach relies on gene similarity; it is based on evidence that similar phenotypes are caused by genes with similar functions. Our algorithm consists of (i) inferring several models (based on various genomic data sources) from a training set of genes, (ii) applying each model to the candidate genes to rank them and, (iii) merging the several rankings into a final ranking of the candidate genes. Recently, we have extended Endeavour to make it a multiple-species tool. Nowadays, the tool supports Homo sapiens, Drosophila melanogaster, Mus musculus, Rattus norvegicus and Caenorhabditis elegans.

As a functional validation, Endeavour was used to optimize a genetic screen performed in Drosophila melanogaster. The goal was to find genes that interact physically with atonal. The regions outputted by the genetic screen were prioritized and the validation showed that Endeavour ranked the true interactors in the top of the regions.

We next applied this concept to heart disorders. Starting from patients with heart defects for which the causal gene is unknown, regions of interest were defined using the array-CGH technology. They were then prioritized in order to find the most promising candidates for further experiments. The first in-situ validations show that Endeavour ranks the best candidates on top, decreasing thus the cost of the validation. In conclusion, we present Endeavour, a framework that can prioritize selected candidate genes or whole genomes in five major organisms, and for which the results were experimentally validated.

Consensus Filtering of Narrow DNA Aberration in SNP Array Data: Proof of Principle

G.Wong [1,2], C. Leckie [1,2], I. Campbell [3], K. Gorringe [3], I. Haviv [1,3] and A. Kowalczyk [1,2]

[1] NICTA, Victoria Research Laboratory, Parkville, Victoria, Australia.
[2] The University of Melbourne, Victoria, Australia.
[3] Peter MacCallum Cancer Centre, East Melbourne, Victoria, Australia.

Motivation: The ability to examine the human genome at high resolution has been enhanced with the introduction of microarray technology with inter-probe distances of less than 1 kilobase in recent releases of the Affymetrix SNP arrays. The ability to identify DNA mutations will assist our understanding of the pathogenesis of cancer, particularly if they are consistent across multiple tumour samples. Narrow regions of change in the human genome often go undetected as algorithms tend to regard individual outlying points as noise and exclude them from the analysis. We address the presence of noise at the sample level with various calibration techniques and compute a set of independent statistics to elucidate consensus change across the genome down to the resolution of a single probe.

Results: Applying our methodology to the Tumour Sequencing Project dataset on lung adenocarcinoma [4], we are able to detect many hundreds of narrow consensus peaks sitting above the Bonferroni-correction threshold. Many identified peaks reside in regions of widely-implicated oncogenes and tumour suppressor genes prompting the need for further biological verification. Our results also show examples of differential peaks between phenotypes, most notably between gender which agrees with known clinic-pathological gender differences in lung cancer. Differences between tumour grades and stages were also examined with results consistent with the characteristics of these phenotypes. The significance of our methodology was independently corroborated by analysis of synthetic data and independent results of more systematic validation using an expanded ovarian cancer dataset form [2].

Conclusion: While other approaches such as GLAD [3] or GISTIC [1] are driven by the amplitudes and frequency of a limited percentage of samples, we take a complementary approach to provide a consensus analysis across all samples in identifying significant narrow regions that are consistently amplified or deleted in the sample space. Our results are largely orthogonal and complementary to all methods for copy number analysis known to us, which often regard micro-regions of change as outliers and exclude them from their analysis. However, we argue that statistically significant micro-regions can be identified from analysis across multiple samples. These micro-regions of aberration could be indicative of concealed biology which may not have otherwise surfaced through the application of other fore mentioned techniques.

References

1 R. Beroukhim, *et al*., *Proc Natl Acad Sci U S A*, 104(50):20007–12, 2007.
2 K. L. Gorringe, *et al*., *Clin Cancer Res*, 13(16):4731–9, 2007.
3 P. Hupe, *et al*., *Bioinformatics*, 20(18):3413–22, 2004.
4 B. A. Weir, *et al*., *Nature*, 450(7171):893–8, 2007.

Comparison of Transfer and Classification Approaches For Function Annotation From Sequences

Ayşe Gül Yaman[1], Ömer Sinan Saraç[1], Rengül Çetin-Atalay[2], and Volkan Atalay[1]

[1] Department of Computer Engineering, Middle East Technical University, Ankara TURKEY,
aysegul,sarac,volkan@ceng.metu.edu.tr
[2] Department of Molecular Biology and Genetics, Bilkent University, Ankara TURKEY
rengul@bilkent.edu.tr

Proteins are essential gene products for living organisms. They have important functions for the continuation of life. Annotating functions of proteins is an important problem for biologists due to the increasing number of identified proteins. The difficulty of manually curating the annotations of proteins necessitate the use of computational methods. Sequence similarity based systems are mostly used by biologists because of the majority of known sequences. There are two approaches used for functional annotation of proteins in the literature. The first approach is based on transferring annotations of homologous proteins according to the results of sequence similarity search in a database of proteins with known annotations. We call this method as transfer approach. The second method is the classification approach treating the functions as classes and proteins as the samples to be classified. In this method the positive and negative training datasets of a class are used for training the classifier and this classifier is used to determine the annotation of the query protein. Our aim in this study is to make an accurate comparison between these two approaches using specific organisms. We use 9 model organisms for our experiments. We select 300 molecular function terms from Gene Ontology (GO). We extract the proteins of model organisms annotated with these 300 terms from the Gene Ontology annotation file and their sequences from UNIPROT database. For the transfer approach we prepare a database of proteins from 8 organisms as the training data and use the proteins of the remaining organism as the test data. For each protein in the test organism we search the database by using BLAST and transfer the annotations of the proteins which have e-values below a specific threshold. We compare these annotations with real annotations of the protein and assess these comparisons based on sensitivity and specificity values. For the classification approach we prepare positive and negative datasets for each term considering the directed acyclic graph structure of GO. 8 of the organisms are used to generate the training set and one organism is used to generate the test set. We apply BLAST-kNN(k nearest neighbor) algorithm to the training dataset, classify the proteins in the test organism and calculate the sensitivity and specificity values. BLAST-kNN is selected in order to make a fair comparison with transfer approach by using BLAST for both approaches.

References

1. O.S. Sarac, O. Gursoy-Yuzugullu, R. Cetin-Atalay, V. Atalay, "Subsequence based feature map for protein function classification", Journal of Computational Biology and Chemistry, Elsevier, Vol.32, pp.122-130, 2008
2. Omer Sinan Sarac, Rengul Cetin-Atalay and Volkan Atalay. GOPred: Combining classifiers on the GO , International Workshop on Machine Learning in Systems Biology, 2008
3. Gaurav Pandey, Vipin Kumar and Michael Steinbach, "Computational Approaches for Protein Function Prediction: A Survey", TR 06-028, Department of Computer Science and Engineering, University of Minnesota, Twin Cities [Bibtex]

Clustering genes by genomic data fusion

Shi Yu, Leon-Charles Tranchevent, Bart De Moor and Yves Moreau

Bioinformatics group, SCD, Department of Electrical Engineering,
Katholieke Universiteit Leuven,
Kasteelpark Arenberg 10, B-3001 Heverlee, Belgium

Two trends underlie the work described in this poster. First, while clustering genes based on microarray expression data is a staple of computational biology, how to cluster genes based on multiple types of heterogeneous data remains a largely open question. Second, data integration across heterogeneous data types using kernel methods have been recently proposed for supervised learning and novelty detection. However, few clustering methods have yet been developed within the same framework. In real-world problems, clustering problem often faces with diverse information obtained from multiple measurements or heterogeneous data sources. The idea of combining information from heterogeneous genomic data sources to endow clustering algorithm with the ability to retrieve similar or complementary information about the underlying partitions of the same set of objects has started to attract interest. The key motivation behind our work is that kernel methods provide a principled framework to integrate multiple types of heterogeneous data - an approach we call genomic data fusion. We propose here a novel adaptive kernel K-means clustering algorithm that goes beyond our previous work on supervised learning and novelty detection. The main advantage of the proposed method is that it fuse the information from heterogeneous data sources as a weighted combinations of similarity matrices (and thus of kernel matrices). By going through a representation of the data as similarity matrices, we "isolate" ourselves from the fact that the different data are often vastly different as to their dimensionality and the type of features that describe their data (be it vector data, sequences or even graphs). Because the kernels are all represented in matrices of the same size, they can be - in a first instance - combined straightforwardly and elegantly in a linear model with uniform weights (i.e., averaging the kernels). Obviously, different data sources can have a different relevance to the problem at hand, so that we want to go beyond a simple average of kernels. In a first step, only a subset of data sources can be selected manually based on knowledge of the data integration problem at hand (i.e., which data sources are likely to be informative based on expert knowledge). A more satisfactory approach is to carry out the combination of kernels through a weighted combination of the individual kernels and to determine those weights through machine learning and optimization.

As our main application domain is the discovery of human disease-causing genes, our method is assessed against a benchmark of clustering human disease genes and compared to several other clustering strategies. For this problem, the proposed method outperforms other methods and performs better than any clustering result obtained on any single type of data; hereby demonstrating the effectiveness of our clustering strategy and the usefulness of data fusion.

Kernel ENDEAVOUR: a web platform for kernel based gene prioritization with data fusion

Shi Yu[1] , Leon-Charles Tranchevent[1] , Roland Barriot[1], Daniela Nitsch[1], Tijl De Bie[3], Bart De Moor[1] and Yves Moreau[1]

[1] Bioinformatics group, SCD, Department of Electrical Engineering, Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, B-3001 Heverlee, Belgium
[2] Department of Engineering Mathematics, University of Bristol, University Walk, BS8 1TR, Bristol, UK

Endeavour is a web platform for the prioritization of genes powered by genomic data fusion. Kernel Endeavour is a latest update version which combines Endeavour with kernel methods. In Kernel Endeavour, gene prioritization is regarded as a kernel based novelty detection problem and information from multiple data sources is integrated as a weighted combination of kernels. It has been shown in previous research that Kernel Endeavour has better performance than Endeavour in disease based genes validation (De Bie et al., 2007).

Kernel Endeavour avoid the difficult parameter tuning problem in kernel construction by adaptive fusion of a series of pre-computed kernels. In our implementation, each data source has 5 pre-computed kernels (1 linear kernel and 4 RBF kernels). The parameter of RBF kernels are selected empirically. The key strength of Kernel Endeavour is that it can combine a series of kernels and the prioritization algorithm is capable to weigh these kernels adaptively. This adaptivity in data fusion reinforces the robustness to noisy information, which might represented by some kernels constructed by bad parameters. This adaptivity also guarantee that if the series of input kernels do not contain the optimal kernel but does contain some suboptimal kernels, the final prioritization result after data fusion is still reliable.

Another challenge of Kernel Endeavour is that the construction of kernel matrices representing the whole genome information is very computational expensive. In order to tackle this problem, we apply incomplete cholesky decomposition (ICD) to reduce the dimensionalities of kernel matrices. The original gene expression data is retrieved from Endeavour database and the complete kernel matrix of the whole genomic data is computed. Then, we apply ICD on these full-size kernel matrices and obtain side matrices of smaller size. The kernel matrices in run-time are reconstructed as the scalar product of these reduce-size side matrices. The overall application is separated into online and offline parts. The offline application handles all the computational heavy procedures such as kernel computation, ICD. The online application loads the decomposed side matrices, retrieves information about the relevant genes and reconstructs the corresponding kernel matrices. The optimization problem of kernel based prioritization is solved by MOSEK toolbox. Due to this online-offline separation, Kernel Endeavour is able to give user an efficient response in genome-wide prioritization.

Multi-spectral biclustering for data described by multiple similarities

Farida Zehraoui*, Florence d'Alché-Buc*,++
*IBISC CNRS fre 3190, Université d'Evry-Val d'Essonne & Genopole, FRANCE
++ URA CNRS 2171, Institut Pasteur, Paris, France.
Emails: farida.zehraoui, florence.dalche@ibisc.fr

In computational biology, objects of interest such as proteins or genes can be described from various points of view as sequences, trees, nodes in a graph, vectors... Often only similarity matrices are available to represent each of these heterogeneous views. Investigating the relationships among these data is an important step toward understanding the biological functions.
Existing data mining approaches, which deal with heterogeneous data, aim to extract objects that are similar among all the views. As the number of datasets increases, it is often not possible to find subsets of objects simultaneously similar according all the views, except in trivial cases.
We thus propose an extension of biclustering, called *multi-spectral biclustering*, that allows to find subgroups of objects that are similar to each other according some of the views. The new algorithm is based on multiple low dimensional embeddings of the data using Laplacian of graphs weighted by the various similarities and a generalization of the squared residue minimization biclustering algorithm ([1,5]). We also propose to select biclustering parameters using a stability criterion [2].
We have sucessuflly tested *muti-spectral biclustering* on two biological applications and obtained very good results. The first application concerns two classes of proteins (membrane proteins and ribosomal proteins) described by several data sets (the protein sequences, the hydropathy profiles of the proteins, etc.) [4]. The second one deals with yeast time series genes expressions measured in several conditions differing by the kind of the strain (wt, mec1, dun1) and the type of stress (MMS, Gamma, mock) [3].

References

[1] Belkin, M. and Niyogi, P. Laplacian eigenmaps and spectral techniques for embedding and clustering. Advances in neural information processing systems, 14, 585-591 (2002).
[2] Ben-Hur, A. Elisseeff, A. and Guyon, I. A stability based method for discovering structure in clustered data, Pac Symp Biocomput. 7, 6-17 (2002).
[3] Gasch, A.P. , Huang, M. , Metzner, S., Botstein, D. Elledge, S.J. and Brown P.O. Genomic expression responses to DNA-damaging agents and the regulatory role of the yeast ATR homolog Mec1p. Mol Biol Cell,vol. 12, 2987-3003 (2001).
[4] Lanckriet, G-R.G., De Bie, T., Cristianini, N., Jordan M.I. and Noble, W.S., A statistical framework for genomic data fusion, Bioinformatics, 20(16), 2626—2635 (2004).
[5] Sra, S., Cho, H., Dhillon I.S. and Guan, Y. Minimum Sum-Squared Residue Co-Clustering of Gene Expression Data. SDM (2004).