Edited by Pekka Ruusuvuori, Tiina Manninen, Heikki Huttunen, Marja-Leena Linne and Olli Yli-Harja

Proceedings of

The 4th TICSP Workshop on Computational Systems Biology, WCSB 2006

Tampere 2006
2. Yaroslavsky. Target Location: Accuracy, Reliability and Optimal Adaptive Filters.
31. Astola, Danielian. Frequency Distributions in Biomolecular Systems and Growing Networks

Tampere International Center for Signal Processing
Tampere University of Technology
P.O. Box 553, FI-33101 Tampere, Finland

ISSN 1456-2774

Juvenes Print - TTY, 2006
TTY:n kuva-arkisto
PREFACE

The Workshop on Computational Systems Biology is an annual two-day event organized by the Tampere University of Technology (TUT). The history of the workshop traces back to 2003, when it was organized for the first time as an internal meeting with some of invited collaborators. Since then the meeting has grown each year witnessing a need for a forum utilizing both sides of systems biology research: there has been a rapid increase in the experimental wet-lab based research and in the research of computational methods in systems biology.

This year the program committee set the target to increase the length and significance of the research papers published in this proceedings book. The maximum length of the papers was increased from two pages to four pages allowing authors to write full length research papers. We received four-page papers from eight different countries, which makes the workshop more international than ever. This is partly to thank the EU IST Biopattern project which also holds its executive group meeting in Tampere at the same time.

In the call for papers, four topics were mentioned: Applications of Image Processing in Systems Biology, Discrete Models and Ensemble Approach in Theoretical Biology, Simulation, Modeling and Data Analysis in Systems Biology and Applications of Computational Methods in Cell and Molecular Biology. Of these, Simulation, Modeling and Data Analysis received the highest number of contributions; more than half of the papers are classified under this topic. This indicates that the tremendous growth in high-throughput methods has resulted in large datasets in systems biology and therefore created a need for sophisticated analysis and modeling tools that can be used to interpret complex biological phenomena.

This volume is the collection of the research papers and short abstracts submitted to WCSB2006. We would like to thank the authors and the reviewers for their contributions to this workshop. We are also grateful for the members of the systems biology group at TUT and especially those who participated in the local organization. We also thank the Finnish Academy of Sciences and Tampere Graduate School in Information Science and Engineering (TISE) and Tampere International Center for Signal Processing (TICSP) for their support.

Heikki Huttunen  Marja-Leena Linne  Olli Yli-Harja
Lecturer  Academy Research Fellow  Professor
Institute of Signal Processing
Tampere University of Technology
Tampere, Finland
# TABLE OF CONTENTS

## Abstracts

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robustness and Evolvability in Genetic Regulatory Networks</td>
<td>3</td>
</tr>
<tr>
<td>Maximino Aldana-Gonzalez, Universidad Nacional Autonoma de Mexico, Mexico</td>
<td></td>
</tr>
<tr>
<td>Mean Field Model of Genetic Regulatory Networks</td>
<td>5</td>
</tr>
<tr>
<td>Mireea Andrecut, University of Calgary, Canada</td>
<td></td>
</tr>
<tr>
<td>Enhancing &quot;New Science&quot; Outcomes of the Policy Process: Some Philosophical Problems</td>
<td>7</td>
</tr>
<tr>
<td>Robert A. Este, University of Calgary, Canada</td>
<td></td>
</tr>
<tr>
<td>Automated Quantitative Analysis of Biomedical Microscopy Images</td>
<td>9</td>
</tr>
<tr>
<td>Antti Niemistö, Tampere University of Technology, Finland</td>
<td></td>
</tr>
<tr>
<td>Modelling and Simulation of Biochemical Pathways</td>
<td>11</td>
</tr>
<tr>
<td>Sam Roberts, The MathWorks, UK</td>
<td></td>
</tr>
<tr>
<td>Biological Data Integration — Get into the 'Do' Mode</td>
<td>13</td>
</tr>
<tr>
<td>Christophe Roos, MediCel Ltd., Finland</td>
<td></td>
</tr>
<tr>
<td>Analysis of Morphome Information</td>
<td>15</td>
</tr>
<tr>
<td>Maria Samsonova, St. Petersburg State Polytechnical University, Russia</td>
<td></td>
</tr>
<tr>
<td>Advances in Data Integration and Representation in Systems Biology</td>
<td>17</td>
</tr>
<tr>
<td>Susie Stephens, Oracle</td>
<td></td>
</tr>
<tr>
<td>Stein-Type Regularized Inference for Complex Biological Models</td>
<td>19</td>
</tr>
<tr>
<td>Korbinian Strimmer, University of Munich, Germany</td>
<td></td>
</tr>
</tbody>
</table>

## Regular papers

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finding Infeasible Reactions in Metabolic Networks: Comparison of Methods</td>
<td>23</td>
</tr>
<tr>
<td>Tommi Aho, Tampere University of Technology, Finland; Thomas Wilhelmi, Andreas Beyer, Leibniz Institute for Age Research - Fritz Lipmann Institute, Germany; Stefan Schuster, Friedrich-Schiller-University Jena, Germany; Olli Yli-Harja, Tampere University of Technology, Finland</td>
<td></td>
</tr>
<tr>
<td>Multilayer Neural Network Based on Multi-Valued Neurons (MLMVN) Applied to Classification of Microarray Gene Expression Data</td>
<td>27</td>
</tr>
<tr>
<td>Igor Aizenberg, Texas A&amp;M University-Texarkana, USA; Pekka Ruusuvuori, Olli Yli-Harja, Jaakko T. Astola, Tampere University of Technology, Finland</td>
<td></td>
</tr>
<tr>
<td>Redefinition of Probe Sets Improves the Comparability of the Data between Affymetrix Array Generations</td>
<td>31</td>
</tr>
<tr>
<td>Reija Autio, Tampere University of Technology, Finland; Sami Kilpinnen, VTT Technical Research Centre of Finland, University of Turku, University of Helsinki, Finland; Sampsa Hautaniemi, Tampere University of Technology, Finland; Olli Kallioniemi, VTT Technical Research Centre of Finland, University of Turku, Finland; Jaakko Astola, Tampere University of Technology, Finland</td>
<td></td>
</tr>
<tr>
<td>EEG Preprocessing by Source Derivation for User Defined EEG Montage and Reference Conversion of Standard Recordings and Evoked Potentials</td>
<td>35</td>
</tr>
<tr>
<td>Cristin Bigan, Ecological University of Bucharest, Romania; Mirea Besleaga, Ecological University of Bucharest, Romanian Society for Clinical Neurophysiology, Romania</td>
<td></td>
</tr>
</tbody>
</table>
A Model-Based Approach to Capture Genetic Variation for Future Association Studies
Susana Eyheramendy, Jonathan Marchini, Gideon McVean, University of Oxford, UK; Simon Myers, Broad Institute of MIT and Harvard, USA; Peter Donnelly, University of Oxford, UK

Statistical Conservation Analysis of Zinc-Interacting Residues
Ioannis N. Kasampalis, Ioannis Pitas, Kleoniki Lyroudia, Aristotle University of Thessaloniki, Greece

Bayesian Orthogonal Least Squares (BOLS) Algorithm for Reverse-Engineering Gene Regulatory Networks
Chang Sik Kim, University of Tampere, Finland; Tapio Salakoski, University of Turku, Finland; Mauno Vihinen, University of Tampere, Finland

Modeling of Rate-Dependent Regulation in Excitation-Contraction Coupling of Mouse Ventricular Myocyte
Jussi Koivumäki, Topi Korhonen, Pasi Tavi, Matti Weckström, University of Oulu, Finland

Gene Expression Predictors of Breast Outcome: A Visualisation Study
David Lowe, Mingmanas Sivaraksa, Aston University, UK

Hierarchical Neuroscope – An Approach to Gene Expression Data Visualisation
David Lowe, Xunxian Wang, Ian Nabney, Aston University, UK

Simulation Study of Deterministic Differential Equation Model for Protein Kinase C Signaling: Sensitivity of Stimuli, Parameter Values, and Initial Concentrations
Tiina Manninen, Antti Saarinen, Keijo Ruohonen, Marja-Leena Linne, Tampere University of Technology, Finland

Extraction of Quantitative Gene Expression Data from the Images of Gene Expression Patterns with Prostack and ISIMBioS
Anna Matveeva, Konstantin Kozlov, Maria Samsonova, St. Petersburg State Polytechnical University, Russia

Analyzing Boolean Network Dynamics using Attractor Basin Structure
Matti Nykter, Juha Kesseli, Tampere University of Technology, Finland; Ilya Shmulevich, Institute of Systems Biology, USA; Olli Yli-Harja, Tampere University of Technology, Finland

Using Regularized Dynamic Correlation to Infer Gene Dependency Networks from Time-Series Microarray Data
Rainer Opgen-Rhein, Korbinian Strimmer, University of Munich, Germany

Immunome: Collection of Genes and Proteins for Systems Biological Studies on Human Immune System
Csaba Ortutay, Markku Siemrma, University of Tampere, Finland; Mauno Vihinen, University of Tampere, Tampere University Hospital, Finland

Self-Overlap in Boolean Networks
Pauli Rämö, Juha Kesseli, Olli Yli-Harja, Tampere University of Technology, Finland

Parameter Estimation for Hodgkin-Huxley Type of Models
Antti Saarinen, Marja-Leena Linne, Olli Yli-Harja, Tampere University of Technology, Finland

Automated Analysis of Golgi Apparatus Dispersion in Neuronal Cell Images
Jyrki Selinummi, Antti Lehmussola, Tampere University of Technology, Finland; Jertta-Riina Sarkanen, University of Tampere, Finland; Jonna Nyky, Tuula O. Jalonen, University of Jyväskylä, Finland; Olli Yli-Harja, Tampere University of Technology, Finland

On the Distribution of Small Avalanches in Random Boolean Networks
Roberto Serra, Marco Villani, Università di Modena e Reggio Emilia, Italy; Stuart A. Kauffman, University of Calgary, Canada
ABSTRACTS
Living organisms are robust to a myriad of random perturbations, both internal and external. Gene regulation networks and metabolic pathways self-organize and reaccommodate to make the organism perform with stability and reliability under perturbations. At the same time, living organisms are evolvable, which means that internal perturbations can eventually make the organism acquire new functions and adapt to new environments. It is still an open problem to determine how robustness and evolvability blend together to produce stable organisms that yet can change and evolve. Here we address this problem by studying the dynamical stability of genetic regulatory network models under the process of gene duplication and divergence. We show that an intrinsic property of this kind of networks is that, after the divergence of the parent and duplicate genes, with a high probability the previous functions of the network are preserved and new ones might appear. The above is true in a variety of network topologies and even for the case of extreme divergence in which the duplicate gene bears no relation with its parent. Consequently, the robustness observed in the network dynamics is not associated with any kind of gene redundancy. Rather, it seems to be a distributed robustness produced by the collective behavior of the entire network.
We discuss the dynamics of a mean field model of the genetic regulatory network. We show that depending on the set of regulatory parameters, the model exhibits complex behavior corresponding to regular and chaotic dynamics, including an order-chaos transition. A more detailed analysis shows that the complex solutions are actually confined in a finite interval corresponding to a relatively small number of interactions per gene. This interval consists in a region of relative stability between two maxima of complex behavior. Therefore, we may say that this region corresponds to a critical phase where the most complex coordinated behaviors can occur. The model also provides a possible explanation of how diversity, stability and robustness are created in a biological system, giving rise to a great variety of stable living organisms.
In this paper we assert that the emergence of "new science" is a specific example of the general case of consequences of the policy process. To commence support of this assertion we briefly examine Holonic Systems in AI, aspects of Quantum Mechanics, and especially, Systems Biology. We explore how philosophical problems commonly associated with the emergence of "new science" are related to this process. We then seek to determine if such philosophical problems can be addressed so that the policy process can be improved, and the emergence of "new science" enhanced.

We employ the "standard model" of the policy process in our analysis, where the technical, political, and conceptual components of that process are dynamically arranged. We reveal the distribution of policy process components in action and explore why these components appear to have a common distribution. We then explore how this common distribution affects the outcomes of the policy process, and ask whether plausible solutions to the aforementioned philosophical problems might allow us to re-engineer this distribution in order to enhance the emergence of "new science".

We close by suggesting that further philosophical analysis of the policy process may have considerable positive value in terms of advancing science policy in general, and science policy related to the emergence of Systems Biology in particular.
Traditionally biological samples have been analyzed manually by visual inspection under the microscope. For example, a simple task could be to count the number of cells in a cell population. Analysis performed in this way is naturally very labor intensive, tedious, and slow. Moreover, if a quantitative manual analysis is made by two different persons, the results may not be the same. This is known as inter-observer variability. Intra-observer variability can also be observed, that is, when the same person performs the analysis twice, the results may differ. If a digital camera is attached to the microscope and digital images of the biological samples are obtained, automated image analysis can be used to overcome all of the above mentioned problems. The same criteria and algorithms are always used in detecting and quantifying the desired features from the images, and the analysis is always performed objectively. Since the analysis can always be performed in exactly the same way, results obtained with automated image analysis are also reproducible. Although the solution to an image analysis problem depends heavily on the nature of the image data, the general image analysis procedure is usually the same. The main steps of the procedure are image acquisition, image pre-processing, image segmentation, feature extraction, validation, and data analysis. In the talk, these steps are described with the aid of illustrative example images.
MODELLING AND SIMULATION OF BIOCHEMICAL PATHWAYS

Sam Roberts
The MathWorks, U.K.
sam.roberts@mathworks.co.uk

We will introduce SimBiology, a new platform for modelling, simulating and analysing biochemical pathways. SimBiology offers the ability to graphically or programmatically create models, combined with industry-proven simulation and analysis techniques, such as parameter estimation and sensitivity analysis. SimBiology is integrated with the MATLAB platform, enabling SimBiology to leverage MATLAB algorithms, data analysis and visualisation. We will demonstrate the use of SimBiology to model and analyse a Yeast Heterotrimeric G Protein Cycle model.
BIOLOGICAL DATA INTEGRATION – GET INTO THE ’DO’ MODE

Christophe Roos
Systems Biology Platform Development
MediCel Ltd., Helsinki, Finland
christophe.roos@medicel.com

System level information can now be collected by measuring almost any component of a biological organism. The multiple fields of ’omics’ have proven this for some years already. To some extent this opportunity has reversed the scientific process whereby formerly an experiment was designed to answer a question while now, the questions tend to come after the measurements have been made. As unprecedented amounts of data are produced by the use of high throughput technologies, the intellectual process of making sense out of the data is quite a challenge.

Data integration on the conceptual level has quite some time been recognised as a key issue carrying huge benefits. Nevertheless, integration and the ensuing process of knowledge extraction are still hampered by a lack of standardised representation of the data in use. A more clear distinction between component data, system data and state data also needs to be established. In this presentation, the crucial and tight interplay between information technology and biology is set as the requirement sine qua non for getting into the ’do integration’ mode.
ANALYSIS OF MORPHOME INFORMATION

Maria Samsonova

Department of Computational Biology
St. Petersburg State Polytechnical University, St. Petersburg, Russia
samson@spbcas.ru

To characterize the development in the morphogenetic field controlling Drosophila segmentation we take a systems biology approach by combining high-precision assay of gene expression with statistical analysis and mathematical modeling [1-3]. For several years we have worked at testing the hypothesis that segmental architecture is determined by a regulatory cascade of transcription factors expressed from segmentation genes. This hypothesis is assumed to be true by Drosophila research community, and is consonant with the results of our own earlier work [3,4]. However, recently the generality of this idea has been called into a question by a study [5] demonstrating that the accuracy of segment determination may not directly controlled by the regulatory cascade of segmentation genes. This result is itself a special instance of a general phenomenon, evident in our own data, in which highly variable early expression patterns become extremely uniform across embryos when the time of critical determination events arrives. In this talk I will discuss the mechanism and precision of pattern formation, as well as buffering mechanisms compensating for variation in segmentation gene expression.

References


ADVANCES IN DATA INTEGRATION AND REPRESENTATION IN SYSTEMS BIOLOGY

Susie Stephens
Principal Product Manager
Life Sciences, Oracle
susie.stephens@oracle.com

A core requirement of advancing systems biology understanding is the integration of data sets. This is a result of biology developing into a "big" science, where different research groups focus on solving different parts of the biological puzzle. To date, the integration of data has been challenging to achieve due to large, complex data sets, which have embraced different data types, terminology, and identifiers. There is frequently also heterogeneity in the biological content of the data. This presentation focuses on recent advances in the integration and representation of heterogeneous systems biology data.
Understanding complex biological networks on a whole-genome scale is a central objective of systems biology. However, the increasing post-genomic information flood offers substantial challenges for the systems analysis of genomic data.

In my talk I focus on methodological problems related to modeling, inferring and simulation of complex networked systems. A key issue is the fit of high-dimensional models with many parameters (which correspond to genes, kinetic parameters, network edges, etc.) to genomic data that are typically are sampled from only few individuals.

In order to deal with this "small n, large p" data situation we have developed an approach to Stein-type shrinkage estimation for the complex high-dimensional models encountered in systems biology. This procedure is computationally very cheap (in comparison to regularized inference based on as penalized likelihood or Bayesian procedures) and thus is ideal for the large genomic and proteomic data sets. Nevertheless, the proposed approach is statistically highly efficient.

Specifically, we have applied this method to infer large scale linear graphical models, such as graphical Gaussian models, structural equations models, and vector autoregressive models from gene expression data, to describe the network-like dependencies among genes.
REGULAR PAPERS
FINDING INFEASIBLE REACTIONS IN METABOLIC NETWORKS: COMPARISON OF METHODS

Tommi Aho¹, Thomas Wilhelm², Andreas Beyer², Stefan Schuster³ and Olli Yli-Harja¹

¹Institute of Signal Processing, Tampere University of Technology, P.O. Box 553, FI-33101 Tampere, Finland
²Leibniz Institute for Age Research - Fritz Lipmann Institute, Beutenbergstr. 11, D-07745 Jena, Germany
³Department of Bioinformatics, Friedrich-Schiller-University Jena Ernst-Abbe-Platz 2, D-07743 Jena, Germany
tommi.aho@tut.fi, thomas.wilhelm@fli-leibniz.de, andreas.beyer@fli-leibniz.de, schuster@minet.uni-jena.de, olli.yli-harja@tut.fi

ABSTRACT

Metabolic networks may have reactions which cannot be active when the network is operating in a steady state. Especially, after some reactions are inactivated, there may be a set of other reactions which become infeasible. Various computational methods have been proposed for identifying the infeasible reactions in metabolic networks: Boolean networks, flux balance analysis, and elementary flux modes. In this paper, we explore the correctness and computational efficiency of these methods and present an improvement to the Boolean network based method.

1. INTRODUCTION

Metabolism is of high importance in modern biotechnological research and industrial production. Nonfunctional metabolism is related to various diseases. Industry exploits the metabolic capabilities of cells when e.g. antibiotics are produced.

A metabolic network is in a steady state if there is no accumulation or depletion of any intermediates. The importance of the steady state is that it maintains stable intracellular conditions while allowing constant fluxes of material through the network. Normally cells are able to reach different steady states which are needed e.g. in different stages of its life cycle and when a changing environment requires adaptation.

Metabolic networks can be manipulated for various reasons, such as to achieve better yield in an industrial bioprocess or to prevent the production of a harmful product. One method to modify a network is to delete a gene coding for an enzyme which leads to inactivation of the corresponding enzymatic reaction.

Usually the inactivation of a reaction affects also other parts of the metabolic network. The simplest example is a linear pathway where an enzyme deletion would lead to accumulation of substrates in the upstream pathway (see Fig. 1a and b). However, the task becomes more difficult if somewhat more complex reaction networks are considered (Fig. 1c).

It is possible to identify infeasible reactions in a metabolic network using information of the network structure and the steady state constraint. Several methods are proposed for this task: flux balance analysis [1, 2], elementary flux modes [3,4], and Boolean networks [5,6]. Next, we examine their ability to find the infeasible reactions correctly as well as their computational efficiency.

Figure 1. Three example pathways used throughout the paper. For simplicity, all the stoichiometries are assumed to be one. a) Inactivation of reaction R2 results in accumulation of metabolite A and the lack of substrate of reaction R3. In this case, reactions R1 and R3 are infeasible, i.e. they cannot be active in any steady state. b) Inactivation of any of the reactions makes the other reactions infeasible although reactions R2 and R3 are parallel. c) Inactivation of R2 results in infeasibility of reaction R3, because R4 has to consume all the produced A.
2. IDENTIFYING THE INFEASIBLE REACTIONS

Steady state is the central concept for the identification of infeasible reactions in metabolic networks. In a steady state, the concentrations $c$ of the internal metabolites are required to remain constant in time. Mathematically,

$$\frac{dc}{dt} = Sv = 0$$

where $S$ is a $m$ times $n$ stoichiometric matrix of $m$ metabolites and $n$ reactions, and $v = [v_{rev} v_{irr}]^T$ is a vector of reaction rates of reversible reactions and irreversible reactions, respectively. Furthermore, it is required that the reactions work in their thermodynamically feasible directions, i.e., the rates of the reversible reactions may have arbitrary values, $v_i = R$, $i \in rev$, while the rates of the irreversible reactions may have only non-negative values, $v_i \geq 0$, $i \in irr$.

Reaction $j$ is infeasible if it cannot be active in any steady state. In that case, $v_j = 0$ for all the possible solutions of Eq. 1. On the other hand, reaction $j$ is feasible if it is active in some steady state, i.e., $v_j \neq 0$ for some solution of Eq. 1.

2.1. Flux Balance Analysis

Flux balance analysis (FBA) can be used to identify the infeasible reactions [1, 2]. The feasibility of reaction $j$ is examined by setting up an optimization problem in which the aim is to maximize $v_j$. In the problem, stoichiometries are used as constraints and irreversible reaction rates are bounded to be non-negative.

$$\max v_j$$

such that

$$Sv = 0$$

$$v_i \in R, i \in rev$$

$$v_i \geq 0, i \in irr$$

The problem formulated in Eq. 2 is a standard linear programming problem and it can be solved using common methods derived for this kind of problems, such as the simplex algorithm [7].

Irreversible reactions are infeasible if their maximum rate is zero. In contrast, reversible reactions for which the maximum equals zero may be still feasible because their backward direction may be feasible. For them, the problem in Eq. 2 is solved again, but now as a minimization problem. If also the minimum equals to zero, then the reaction is infeasible.

2.2. Elementary Flux Modes

An elementary flux mode (EFM) describes a flux distribution with relative reaction rates (i.e. $e = [r_1, ..., r_n]^T$, where $e$ is the EFM and $r$'s are the relative reaction rates. $e$ corresponds to $v$ in Eq. 1). EFMs represent a special set of flux distributions because all the possible steady states can be described by their linear combinations [3]. Thus,

$$\frac{dc}{dt} = Sv = SEb = 0$$

where all the $N$ elementary flux modes are included in an EFM matrix $E = [e_1, ..., e_N]$ from which $b$ produces the linear combinations.

**Corollary 1.** If $r_j = 0$ for some reaction $j$ in all the elementary flux modes, then reaction $j$ cannot have a non-zero rate in any steady state flux distribution, i.e. reaction $j$ is infeasible.

An elementary flux mode is defined as follows:

1. EFM fulfills the steady state constraint for the intermediate metabolites (Eq. 1)
2. EFM fulfills the non-negativity constraints for the irreversible reactions
3. EFM is elementary in the sense that it has no proper subset of reactions which could fulfill the steady state constraint alone

**Corollary 2.** If any one active reaction is inactivated in EFM $e$, then the EFM is cancelled, i.e. $e = 0$.

The elementary flux modes can be reasoned easily for small networks. For example, in Fig. 1a there is one EFM which contains all the three reactions. In the EFM, all the reactions have the relative rate one, which means that their absolute rates must be equal compared to each other. This EFM can be scaled by any non-negative real number and thus all the steady states of this small network can be described. It is almost equally easy to see that there is only one EFM for the network depicted in Fig. 1b too. Metabolites have constant amounts only if all the reactions proceed at the same relative rate. In both Figures 1a and 1b the inactivation of any of the reactions results to cancellation of the EFM (Corollary 2) and thus inactivation of the other reactions (Corollary 1).

The network presented in Fig. 1c has two EFMs which can be reasoned or, preferably, calculated using special algorithms [8], [9]:

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
</tr>
</thead>
<tbody>
<tr>
<td>EFM1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>EFM2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Based on EFM1 and EFM2 we can deduce that if R1 is inactivated, then both EFMs are cancelled (Corollary 2) and the other reactions are inactivated (Corollary 1). On the other hand, if R2 is inactivated then EFM2 is cancelled and EFM1 remains. Thus R3 has no EFMs in which it is active. R1 and R4 are active in EFM1 and thus they are the only feasible reactions.

2.3. Boolean Networks

Boolean networks (BN) are proposed as a method for finding the infeasible reactions in a metabolic network [5, 6]. In the proposed approach all the metabolites and reactions are represented as nodes having value 0 (metabolite is absent / reaction is inactive) or 1 (metabolite is present / reaction is active). All the nodes have specific rules which determine their values. The rules for metabolites are:

- Metabolite $M$ has value 1 if any of the reactions producing $M$ has value 1. Otherwise $M$ has value 0.
The rules for reactions are:

- Reaction \( R \) has value 1 if all the substrates of \( R \) have value 1. Otherwise \( R \) has value 0.

The feasibility is examined with the help of BN simulations. All the nodes are first set to value 1. After that the value of one of the reaction nodes is permanently switched to 0 which represents the inactivation of the reaction. The switched node causes a series of consecutive switchings of other nodes based on the given rules, until the network stabilizes into a certain state (a point attractor is reached). For example, inactivation of \( R_1 \) in Fig. 1b causes that all the reactions become inactive and all the metabolites become absent.

The BN method fails to identify those infeasible reactions which are located upstream from the inactivated reaction. For example, inactivating reaction \( R_2 \) in Fig. 1b correctly detects metabolite \( C \) as absent and reaction \( R_4 \) as infeasible. However, value 1 remains for reactions \( R_1 \) and \( R_3 \) which results to the accumulation of \( A \) and \( D \). Thus, the steady state constraint of Eq. 1 is not fulfilled and the method fails to identify all the infeasible reactions correctly.

The BN method can be improved by redefining the rules. The new rules approximate the steady state constraint such that if there is an active reaction producing metabolite \( M \), then there must be an active reaction consuming \( M \). The rules are:

- Metabolite \( M \) has value 1 if any of the reactions producing \( M \) has value 1, and if any of the reactions consuming \( M \) has value 1. Otherwise \( M \) has value 0.

- Reaction \( R \) has value 1 if all its substrates have value 1, and if all its products have value 1. Otherwise \( R \) has value 0.

We refer this type of a Boolean network as a steady state Boolean network (SSBN). As an example, Figure 2 presents the rules when they are applied to the system of Figure 1b. If reaction \( R_2 \) is inactivated and the SSBN model is simulated, then both the downstream and the upstream reactions are correctly found infeasible.

The SSBN method performs well both in upstream and downstream directions in a simple pathway. However, in a more difficult situation it cannot identify all the infeasible reactions. Figure 1c illustrates a case in which the method fails. The rules and the simulation of the network are presented in Figure 3. Inactivation of \( R_1 \) makes all the other reactions infeasible which is the correct result. However, inactivation of \( R_2 \) cannot find \( R_3 \) as infeasible.

3. COMPUTATIONAL RESULTS

In this section we study the computational efficiency of flux balance analysis (Section 2.1) and elementary flux modes (Section 2.2). The two Boolean network based methods (Section 2.3) are left without further analysis, because they are not able to identify all the infeasible reactions correctly.
Table 1. Sizes of the test networks and calculation times in the inactivation study (seconds). For EFM the first number is the computation time of EFMs, and the second number is the analysis time of the obtained EFMs. The computation time is not available when EFMs are calculated for the large network, because the computation was aborted after $1 \times 10^5$ seconds.

<table>
<thead>
<tr>
<th>nr of internal metabolites</th>
<th>nr of reactions</th>
<th>Small</th>
<th>Medium</th>
<th>Large</th>
</tr>
</thead>
<tbody>
<tr>
<td>FBA</td>
<td></td>
<td>1 (+1)</td>
<td>408 ($+2 \times 10^3$)</td>
<td>$&gt; 1 \times 10^5$</td>
</tr>
<tr>
<td>EFM</td>
<td></td>
<td>24</td>
<td>110</td>
<td>1149</td>
</tr>
</tbody>
</table>

Three test problems are set up in Matlab environment [10]. 1) A small-sized network representing parts of the amino acid metabolism in \textit{E. coli} [3], 2) a medium size network of central carbon metabolism in \textit{E. coli} [4], and 3) a large size network of yeast \textit{S. cerevisiae} metabolism [1].

Linear programming toolbox Ip_solve [11] is used when the problem is solved with the help of FBA. Program Metatool [9, 12] is used when the problem is solved with the help of EFMs.

The inactivation study is performed similarly for the three test networks. One reaction is inactivated in the network and the feasibility of the other reactions is examined. This is repeated until all the reactions have in turn been inactivated.

Table 1 presents the details of the network sizes and the computation times needed. A computer with Pentium M (1.3 GHz) processor and 1 Gb RAM was used for the computation. The computation time of elementary flux modes is given separate to the time needed for their analysis. This is because the used Matlab code in the analysis part is not optimized to be as fast as possible.

4. DISCUSSION

We have examined four methods which aim to identify the infeasible reactions in a metabolic network. Flux balance analysis and elementary flux modes are able to produce the correct results, because they fulfill the irreversibility and the steady state constraints. Boolean network based methods do not fulfill the steady state constraint and thus they are unable to identify all the infeasible reactions.

For small and medium size networks both FBA and EFM are applicable methods, because their computation times are reasonable. In larger networks the computational work increases rapidly. In our study, FBA was the only method which could produce the correct results for a large network. This is because FBA does not aim to provide such comprehensive information of all steady states as EFM analysis does.

5. REFERENCES


MULTILAYER NEURAL NETWORK BASED ON MULTI-VALUED NEURONS (MLMVN) APPLIED TO CLASSIFICATION OF MICROARRAY GENE EXPRESSION DATA

Igor Aizenberg1, Pekka Ruusuvuori2, Olli Yli-Harja2 and Jaakko T. Astola2

1 Texas A&M University-Texarkana
Department of Computer and Information Sciences
P.O. Box 5518, 2600 N. Robison Rd. Texarkana, Texas 75505 USA,
2Institute of Signal Processing, Tampere University of Technology,
P.O. Box 553, FI-33101 Tampere, Finland,
igor.aizenberg@gmail.com, pekka.ruusuvuori@tut.fi, olli.yli-harja@tut.fi, jaakko.astola@tut.fi

ABSTRACT
Classification of microarray gene expression data is a common problem in bioinformatics. Classification problems with more than two output classes require more attention than the normal binary classification. Here we apply a multilayer neural network based on multi-valued neurons (MLMVN) to the multiclass classification of microarray gene expression data. Two four-class test cases are considered. The results show that MLMVN can be used for classifying microarray data accurately.

1. INTRODUCTION
A multilayer neural network based on multi-valued neurons (MLMVN) has been introduced in [1] and then it has been developed in [2]. This network and its backpropagation learning is comprehensively observed and developed further in [3]. The MLMVN consists of multi-valued neurons (MVN). That is a neuron with complex-valued weights and an activation function, defined as a function of the argument of a weighted sum. MVN is based on the principles of multiple-valued threshold logic over the field of complex numbers. A comprehensive observation of the discrete-valued MVN, its properties and learning is presented in [4]. A continuous-valued MVN and its learning are considered in [1]-[3]. The most important properties and advantage of their learning is that it does not require differentiability of the activation function. The MVN learning algorithm [3], [4] is based on a simple linear error correction rule. This learning rule is generalized for the MLMVN as a backpropagation learning algorithm [3], which is simpler and more efficient than traditional backpropagation learning. MLMVN outperforms a classical multilayer feedforward network (usually referred to as a multilayer perceptron - MLP) and different kernel-based networks in the terms of learning speed, network complexity, and classification/prediction rate tested for such popular benchmark problems as the parity n, the two spirals, the sonar, and the Mackey-Glass time series prediction [1]-[3]. These properties of MLMVN show that it is more flexible and adapts faster in comparison with other solutions based on neural networks. It is important to note that since MLMVN (as well as a single MVN) implements such mappings that are described by multiple-valued (up to infinite-valued) functions, it can be an efficient mean for solving the multiclass classification problems.

In this paper we apply MLMVN to the multiclass classification of microarray gene expression data. After presenting the basic properties of MLMVN and its backpropagation learning algorithm we will consider two four-class test cases of microarray gene expression data classification. The classification results of MLMVN classifier are compared to those given by nearest neighbor classifiers with different numbers of neighbors.

2. MULTILAYER NEURAL NETWORK BASED ON MULTI-VALUED NEURONS

2.1. Multi-valued neuron (MVN)
MVN [4] is a neural element based on the principles of multiple-valued threshold logic over the field of complex numbers. A single MVN performs a mapping between n inputs and a single output. For the discrete-valued MVN this mapping is described by a multiple-valued (k-valued) function of n variables \( f(x_1, \ldots, x_n) \) with \( n+1 \) complex-valued weights as parameters:

\[
 f(x_1, \ldots, x_n) = P(w_0 + w_1 x_1 + \ldots + w_n x_n),
\]

where \( X = (x_1, \ldots, x_n) \) is a vector of inputs (a pattern vector) and \( W = (w_0, w_1, \ldots, w_n) \) is a weighting vector. The inputs and output of the discrete-valued MVN are the \( k^b \) roots of unity: \( e^{j} = \exp(2\pi j/K), \ j = 0,\ldots,k-1, \)
where $i$ is an imaginary unity. $P$ is the activation function of the neuron:
\[ P(z) = \exp(i \arg(z)) \]
where $j = 0, \ldots, k-1$ are the values of $k$-valued logic, $z = w_0 + w_1 x_1 + \ldots + w_n x_n$ is a weighted sum, $\arg(z)$ is the argument of the complex number $z$. Function (2) divides a complex plane onto $k$ equal sectors and maps the whole complex plane into a set of $k$th roots of unity (see Figure 1).

The activation function (2) is discrete. It has been recently proposed in [1]-[3], to modify the function (2) in order to generalize it for the continuous case in the following way. If $k \to \infty$ in (2) then the angle value of the sector (see Figure 1) tends to zero. Hence, the function (2) is transformed in this case as follows:
\[ P(z) = \exp(i \arg(z)) = e^{i \text{Arg } z} = \frac{z}{|z|}, \] (3)
where $\text{Arg } z$ is a main value of the argument of the complex number $z$ and $|z|$ is its modulo. Thus the activation function (3) determines a continuous-valued MVN. Inputs and output of this neuron are lying on the unit circle, but since they are continuous, this case corresponds to the infinite-valued logic.

Figure 1. Geometrical interpretation of the MVN activation function.

It is also possible to consider a hybrid MVN (either discrete inputs $\Rightarrow$ continuous output or continuous inputs $\Rightarrow$ discrete output). We will use in this paper exactly MVN with the continuous inputs and a discrete output. It is important that MVN learning does not depend on type of the neuron. It is reduced to the movement along the unit circle. This movement does not require a derivative of the activation function. The learning process is based on the following error correction rule [3], [4]
\[ W_{r+1} = W_r + \frac{C_r}{(n+1)} (T - Y) \overline{X}, \] (4)
where $X$ is an input vector, $n$ is a number of neuron’s inputs, $\overline{X}$ is a vector with the components complex conjugated to the components of vector $X$, $r$ is the number of iteration, $W_r$ is a current weighting vector, $W_{r+1}$ is a weighting vector after correction, $T$ is a desired neuron's output, $Y$ is an actual neuron's output, and $C_r$ is a learning rate.

2.2. MVN-based Multilayer Feedforward Neural Network (MLMVN)

A multilayer architecture of the network with a feedforward dataflow through nodes that requires full connection between consecutive layers and an idea of a backpropagation learning algorithm was proposed in [5] by D. E. Rumelhart and J. L. McClelland. A classical example of such a network is a multilayer perceptron (MLP) [5], [6]. Its learning is based on the algorithm of error backpropagation. The error is being sequentially distributed form the "right hand" layers to the "left hand" ones. A crucial point of the MLP backpropagation is that the error of each neuron of the network is proportional to the derivative of the activation function. Usually MLP is based on the neurons with the sigmoid activation function [6].

However, it is possible to use different neurons as the basic ones for a network with the feedforward architecture. A multilayer feedforward neural network based on multi-valued neurons (MLMVN) has been recently proposed in [1]-[3]. This network has at least two principal advantages in comparison with an MLP: higher functionality (an MLMVN with the smaller number of hidden neurons outperforms an MLP with the larger number of hidden neurons [1]-[3]) and simplicity of learning (MLMVN learning does not require differentiability of the activation function).

Let us consider $m$-layer MLMVN with $n$ inputs ($m$-1 hidden layers and one output layer (the $m$th one) based on the MVN with the continuous inputs and a discrete output. Let $w_{ij}^k$ be the weight corresponding to the $j$th input of the $k$th neuron ($k$th neuron of the $j$th layer), $y_{ij}$ be the actual output of the $j$th neuron from the $j$th layer ($j = 1, \ldots, m$), and $N_j$ be the number of the neurons in the $j$th layer. It means that the neurons from the $j+1$st layer have exactly $N_j$ inputs. Let $x_1, \ldots, x_n$ be the network inputs. The backpropagation learning algorithm for the MLMVN is described as follows [3].

The global error of the network taken from the $k$th neuron of the $m$th (output) layer is calculated as follows:
\[ \delta_{km} = T_{km} - Y_{km}, \] (5)

The backpropagation of the global errors $\delta_{km}'$ through the network is used (from the $m$th (output) layer to the $m-1$st one, from the $m-1$st one to the $m-2$nd one, …, from the $2$nd one to the $1$st one) in order to express the error of each neuron $\delta_{ij}$, $j = 1, \ldots, m; i = 1, \ldots, N_j$ by means of the global errors $\delta_{km}'$ of the entire network.

The errors of the $m$th (output) layer neurons are:
\[ \delta_{km} = \frac{1}{s_m} \delta_{km}', \] (6)
where $km$ specifies the $k$th neuron of the $m$th (output) layer, $s_m = N_{m-1} + 1$, i.e. the number of all neurons on
the previous layer (layer $m-1$), which the error is backpropagated to) incremented by 1, $\delta_{i}^{m}$ is the global error of the entire network (5) taken from the $k^{th}$ neuron of the $m^{th}$ (output) layer.

The errors of the hidden layers’ neurons are:

$$\delta_{ij} = \frac{1}{s_{j}} \sum_{j=1}^{N_{j}} \delta_{ij}^{s+1} (w_{k}^{s+1})^{-1},$$

where $kj$ specifies the $k^{th}$ neuron of the $j^{th}$ layer ($j=1,\ldots,m-1$); $s_{j} = N_{j+1} + 1$, $j = 2,\ldots,m$; $s_{1} = 1$ is the number of all neurons on the layer $j-1$ (the previous layer $j$ which error is backpropagated to) incremented by 1. The weights for all neurons of the network are corrected after calculation of the errors. In order to do this, the learning rule (4) is used. Hence, the following correction rules are used for the weights [3]:

$$w_{ij}^{N} = w_{ij}^{N-1} + \frac{C_{N}}{(N_{j}+1)} \delta_{ij}^{N} \frac{1}{1}, i = 1,\ldots,n,$$

$$\bar{w}_{0j}^{N} = \bar{w}_{0j}^{N-1} + \frac{C_{j}}{(N_{j}+1)} \delta_{ij}^{j},$$

for the neurons from the $m^{th}$ (output) layer ($k^{th}$ neuron of $m^{th}$ layer),

$$\bar{w}_{ij}^{N} = \bar{w}_{ij}^{N-1} + \frac{C_{j}}{(N_{j}+1)} \delta_{ij}^{N} z_{ij} \frac{1}{1}, i = 1,\ldots,n,$$

$$\bar{w}_{0j}^{N} = \bar{w}_{0j}^{N-1} + \frac{C_{j}}{(N_{j}+1)} \delta_{ij}^{j},$$

for the neurons from the $2^{nd}$ till $m-1^{st}$ layer ($k^{th}$ neuron of the $j^{th}$ layer $j=2,\ldots,m-1$), and

$$\bar{w}_{ij}^{1} = w_{ij}^{1} \frac{C_{i}}{(n+1)} \delta_{i}^{1} z_{ij} \frac{1}{1}, i = 1,\ldots,n,$$

$$\bar{w}_{0j}^{1} = w_{0j}^{1} \frac{C_{j}}{(n+1)} \delta_{ij}^{1},$$

for the neurons of the $1^{st}$ hidden layer, where $C_{j}$ is a constant part of the learning rate (it should be mentioned that in our experiments for all the neurons we took $C_{j} = 1$). The factor $1/z_{ij}$, where $z_{ij}$ is a weighted sum of the $k^{th}$ neuron on the previous learning iteration, is a variable self-adaptive part of the learning rate, which is used only for the hidden neurons, as it is recommended in [3].

In general, the learning process should continue until the following condition is satisfied:

$$E = \frac{1}{N} \sum_{s=1}^{S} \sum_{i=1}^{N} (\delta_{0}^{s})^{2}(W) = \frac{1}{N} \sum_{s=1}^{S} E_{s} \leq \lambda,$$

where $E_{s}$ is a square error of the network for the $s^{th}$ pattern from the learning set ($E_{s} = \sum_{k=1}^{K} (\delta_{0}^{s})^{2}(W)$), $N$ is the number of patterns in the learning set, and $\lambda$ determines the precision of learning. In particular, in the case when $\lambda = 0$ the equation (11) is transformed to $\forall k, \forall s \delta_{0}^{s} = 0$. We will use exactly the last case in our experiments.

3. DATA DESCRIPTION

We use two publicly available microarray gene expression data sets; "Novartis" and "Lung". Both datasets consist of multiple classes. The "Lung" data set includes 197 samples with 419 features (genes) that represent the four known classes. The "Novartis" data set includes 103 samples with 697 features that also represent the four known classes. Though feature selection is left outside the scope of this study, it should be noted that any screening or selection of features that has possibly been done for the data sets prior to our analysis can have significant effect on the result. For a more detailed description of the data sets, see [7]-[9].

Since using MLMVN we have to put the inputs on the unit circle, the gene expression data was not used in classification as such. We used a simple linear transform (see Section 4) to convert the data to the points on the unit circle. Actually this transform simply changes linearly a range of the data and completely preserves the data nature.

We used a $K$-random subsampling with $K=15$ to separate the data on the training and testing sets. Thus $K=15$ training and testing sets have been created. For the "Lung" data set 44 samples of 197 were used for training and the rest 153 ones for testing for the all $K=15$ cases. For the "Novartis" data set 51 samples were used for training and the rest 52 ones for testing for all $K=15$ cases.

4. SIMULATION RESULTS

To test the MLMVN as a classifier for solving the presented problems, we used the network with one hidden layer and one output layer containing the same number of neurons as the number of classes. The best results for both test data sets are shown by the network with 6 neurons on a single hidden layer (any increase of the hidden neurons amount does not improve the results; on the other hand, the results are a bit worse for a smaller amount of the hidden neurons). Thus taking into account that we have in both classification problems exactly 4 classes, the network $n=6 \rightarrow 4$ (where $n$ is the number of inputs) has been used.

We used the MLMVN with the continuous inputs and a discrete output. However, the hidden neurons were continuous-valued, while the output ones combine the continuous inputs with a discrete output. In order to put the original real-valued inputs to the unit circle, their initial range $[a,b]$ was linearly transformed to $[0,6.27]$. Thus, if $t \in [a,b]$, then $\varphi = \left(\frac{t-a}{b-a}\right)6.27 \in [0,6.27]$. If $\varphi \in [0,6.27]$ then it is considered as the argument of the complex number lying on the unit circle.

To form a discrete output of the network, the following approach has been used. Each neuron from the output layer is responsible for the classification of the
samples belonging to one of the four considered classes. This means that each neuron has to recognize patterns only from one of the four classes and to reject all other patterns. Hence the activation function of all the output neurons separates the complex plane onto two semiplanes: the upper one is reserved for recognition of the patterns from a particular class, while the bottom one is reserved for the rejection.

During the learning process we directed the weighted sum to the angles $\pi/2$ in the upper semiplane and $3\pi/2$ in the bottom semiplane. During the learning process the domains $\pi/2 \pm \pi/8$ and $3\pi/2 \pm \pi/8$ were considered as acceptable.

The learning process converges very quickly starting from the random vectors with the real and imaginary parts belonging to $[0, 1]$. It requires 2-3 minutes using a software simulator developed in the Borland Delphi 5 environment on a PC with Pentium IV 3.0 GHz CPU.

To verify the results, as it was mentioned above, a $K$-random subsampling cross validation with $K=15$ has been used for both problems. For the "Novartis" data set there is 96.35% classification rate, and for the "Lung" data set there is 94.32% classification rate. Since the variation of the error for all 15 data splits is very small (0.41 for the "Novartis" data set and 0.39 for the "Lung" data set), this result is very stable. For comparison, the classification results for the "Novartis" data set by using the $k$ nearest neighbors ($k$NN) classifier with $k = 1, 3, 5$ were 97.69%, 97.44%, and 97.31%, respectively. For the "Lung" data set, the classification accuracy for 1NN classifier was 89.80%, for 3NN it was 91.11%, and for 5NN the accuracy was 92.55%. Exactly the same data transformation and subsampling partitions were used for all classifiers.

We can conclude from these results that for the multiclass gene expression data classification problem the MLMVN shows the results that are comparable with the $k$NN classifier. However, due to the shortcomings of cross validation estimators in small sample settings [10], single results do not necessarily provide a reliable basis for comparison between different classification methods, or for drawing direct conclusions on classifier performance. One classifier shows a bit better result for the one data set, another one shows a bit better result for another data set. However, the microarray test cases should be considered as good examples of possible new application areas of the MLMVN.

5. CONCLUSION

A multilayer neural network based on multi-valued neurons (MLMVN) has proven to be a flexible, accurate and fast algorithm for supervised classification. Here the MLMVN classifier is applied to microarray gene expression data classification. The results for two data sets are comparable with the ones obtained with widely used $k$NN classifiers. In the multiclass classification tasks performed for "Novartis" and "Lung" data sets, relatively simple network ($n \rightarrow 6 \rightarrow 4$) provided 96.35% and 94.32% classification rates, respectively.

A possible direction for future research is to continue exploring the performance of MLMVN classifier in the context of microarray gene expression data. A more extensive set of results with different error estimators could provide more information on the accuracy and a more reliable basis for comparison with other classification methods.

6. ACKNOWLEDGMENTS

This work was supported by the Academy of Finland, project No. 213462 (Finnish Centre of Excellence program (2006 - 2011)).

7. REFERENCES


REDEFINITION OF PROBE SETS IMPROVES THE COMPARABILITY OF THE DATA BETWEEN AFFYMETRIX ARRAY GENERATIONS

Reija Autio1*, Sami Kilpinen1,3*, Samps Hautaniemi1, Olli Kallioniemi2 and Jaakko Astola1

1 Institute of Signal Processing, Tampere University of Technology, FINLAND, 2 Medical Biotechnology, VTT Technical Research Centre of Finland and University of Turku, FINLAND, 3 Biomedical Biochip Center and Institute of Biomedicine, University of Helsinki, FINLAND, *equal contribution, reija.autio@tut.fi

ABSTRACT

The number of gene expression microarray experiments is increasing almost exponentially. In this paper we focus on the Affymetrix microarrays that are based on highly consistent and quality-controlled manufacturing technology, but have still been undergoing major design changes over the years. Affymetrix has increased the number of probe sets on their arrays and constantly redesigned the sequences of all the probe sets. As a result, comparability of data between experiments on old and new array generations has become difficult. Our aim was to redefine the probe sets across the different Affymetrix array generations in order to obtain comparable gene expression values. We first linked all the probes to the collection of transcripts at ENSEMBL, and then identified those that had overlapping sequence between different array generations. Already one to four nucleotide overlap between probes on different platforms was found to be sufficient for significantly increasing the comparability of data. As compared to using identical probes that leads to a loss of >50% of the data, our method made it possible to retain a higher number of informative genes between array generations. This was validated by significantly improved correlation coefficients (from 0.65 to 0.74) between a set of 132 leukemia samples measured on both the HG-U95Av2 and HG-U133A generations. This method should facilitate large-scale meta-analyses of the available in silico transcriptomic data.

1. INTRODUCTION

Microarray technology has made it possible to analyze the expression levels of tens of thousands of genes in a single experiment, making this technology indispensable in the exploration of gene regulation and biology in general. Affymetrix oligonucleotide array platform is one of the earliest and most popular microarrays. However, Affymetrix has constantly improved their probe sets, and in each new array generation a large number of new probes and probe sets have been added. Due to these changes the resulting gene expression values between older and younger generations are usually not comparable. The difference is particularly evident, if the values are studied at the gene level, where the effect of probe set values may be cumulative. The correlation coefficients between technical replicates hybridized to the same array generation are typically > 0.9 but if different array generations are probed with the same cDNA, the correlations decrease dramatically. Interestingly, there are only few probe sets whose probes are identical between different array generations, probably since the design for a new array generation is done each time against the human genome sequence with new improved algorithms. It was recently described that gene expression data on identical probes between different platforms are much more comparable than data from other probe sets. Nimgaonkar et al. [1] reported that the reproducibility of gene expression data across generations is high if the probe sets for each gene have been highly similar.

This issue has many faces. Naturally, the most optimized design algorithms need to be used and the most up to date DNA sequence data should be used when designing probes. However, it is impractical to assume that scientists would be willing and able to always reanalyze their old data from previous samples with every new array generation. In order to be able to utilize the data from older array generations, comparability with new array generations is critical. Dai et al. [2] have redefined the probe sets from the older array generations in order to use only probes that are up-to-date and thus obtain more reliable gene values. In another study Hwang et al. [3] presented a method for improving the comparability between array generations by masking out probes that have no overlap between two array generations.

Here, we have studied the effects of these methods separately and also developed a novel EGAGO (ENSEMBL Gene and Array Generation based Overlapping) method that redefines the probe sets, using only probes that are both up-to-date with the most recent DNA sequence data, and overlapping between array generations. This makes it possible to more effectively combine gene-level expression data between the different array generations.

2. METHOD

We have utilized three methods for defining the probes to be used in gene value calculation. 1) The probes of the array generations are mapped to the newest collec-
ation of transcripts by ENSEMBL in order to use only state-of-the-art genome sequence information when calculating probe set values. 2) The probe set values are calculated based on the probes that have overlap between different array generations, and the gene value is then the median of the probe sets within a gene. 3) Our EGAGO-method that requires the overlap with the newest collection of transcripts and also with the different array generations.

2.1. Probes of array generation and newest collection of transcripts

In order to update the probe set values from older array generations based on the current knowledge Dai et al. [2] have introduced a method for redefinition of probe sets. Only the probes that are present in the newest collection of transcript are included in the analysis. The values for genes are now more reliable since they are calculated based on only those probes that are located in transcripts within ENSEMBL [4] genes. Numbers of genes are collected into Table 1.

Table 1. Number of genes in array generations and number of common ones. Affymetrix probe sets are linked to genes using linkage provided by Affymetrix.

<table>
<thead>
<tr>
<th>Affymetrix U95Av2</th>
<th>U133A</th>
<th>Common</th>
</tr>
</thead>
<tbody>
<tr>
<td>8212</td>
<td>11751</td>
<td>8153</td>
</tr>
<tr>
<td>New CDF-files</td>
<td>7833</td>
<td>11527</td>
</tr>
</tbody>
</table>

2.2. Probes overlapping between array generations

To diminish the variation between array generations Hwang et al. [3] suggest a method for re-computation of the probe set values. All the probes that are not having an overlap between array generations are masked from the analysis. Therefore the probe set values are calculated exclusively based on those probes that overlap between the array generations being compared. There is also a possibility to utilize threshold for these overlapped values, in order to limit how many probes at minimum must one probe set have. The overlaps are defined by the "Best Match" criterion provided by Affymetrix for comparisons between array generations. The number of usable probes and probe sets are collected into Table 2.

Table 2. Number of probe pairs and probe sets in array generations HG-U95Av2 and HG-U133A, and overlapping probes between them.

<table>
<thead>
<tr>
<th>all HG-U95Av2</th>
<th>all HG-U133A</th>
<th>Overlap</th>
</tr>
</thead>
<tbody>
<tr>
<td>12625</td>
<td>22277</td>
<td>8142</td>
</tr>
<tr>
<td>204800</td>
<td>253472</td>
<td>83431</td>
</tr>
</tbody>
</table>

2.3. Probes overlapping between array generations and the newest collection of transcripts

In order to obtain reliable and comparable values from the different array generations both of these previous methods should be considered. We introduce an EGAGO method for combining these approaches. There are thousands of common genes between different array generations. However, the values for these genes are calculated based on the probe sets in the array and there is no guarantee that the probes would be measuring the same gene from the same location of the gene. We use a hypothesis that by selecting only probes that have certain amount of nucleotide sequence overlap between two array generations and the newest transcript sequence, the measurements of expression values would be more comparable and reliable.

The actual workflow can be divided in two phases; 1) removing the probe pairs whose perfect match sequence is not part of any of the transcripts of the gene that the probe is supposed to measure, and 2) removing the probes whose perfect match sequences have no overlap between the array generations.

In the first phase we used BLAT [5] to map the sequence of each probe to the newest sequence of transcripts. It is enough to only map the perfect matches of the probes, because the mismatches will have unchangeable correspondence to the perfect matches. All the probes which do not have a unique match in the collection of transcripts are removed.

In the second phase only the probes that have an overlap with a probe in the other array generation under consideration are accepted to the analysis. The probes that do not have any overlap between any other of the probes within the gene in the other array generation are not accepted to the analysis but removed (Figure 1).

In addition, all genes involved in technical controls are removed since they reflect difference in measurement quality between array generations, and not the actual quantitative difference between parallel measurements of biological features.

The set of probes defining the value for gene $G$ in array generation $a$ can be defined with formula:

$$G_{aga} = \left\{ PV_a \mid O(PM_{a}, PM_{b}) \geq k, \{PM_{a}, PM_{b}\} \subseteq T_a, T_b \subseteq G \ \forall i \right\},$$

where $T$ are the transcripts within a gene $G$ and $PV$ is the probe value in the array generation $a$. $PM$ denotes the perfect match nucleotide sequences of the probe pairs in the array generation, and $O$ is the overlap between probes. The maximum number for $k$ is 25 and if $O(PM_{a}, PM_{b}) = 25$, the probes are the same. Now, only probes that have certain amount of nucleotide sequence overlap between two array generations and the transcripts of the gene are measuring the expression values. Gene expression values were then calculated with custom developed MAS5 algorithm [6], [7] for remaining probes. However, any other pre-processing method for expression value calculation can be utilized as well.

In EGAGO-method the probe values are assigned directly to genes, so no centering has to be done for probe set values when more than one probe set is within one gene. Therefore each gene will have exactly one value and all the probes are having the same effect when the gene value is calculated. Hence, the resulted gene values are more reliable.
3. RESULTS

We have utilized this method for the most used Affymetrix array generations HG-U95Av2 and HG-U133A. The method can however be used for any other array generations as well. There are 8153 common genes when mapped directly from the probe sets to the ENSEMBL genes. However, the values for these genes are calculated based on the probe sets in the arrays that differ a lot between the array generations.

In array generation HG-U95Av2 there are 409600 and in HG-U133A 506944 probes. Using the requirement that there is at least one nucleotide long overlap in at least one probe within the ENSEMBL transcripts of the gene, only 198862 (48.6%) probes from HG-U95Av2 and 128576 (25.4%) from HG-U133A can be used. Since the probe pair number is half of the amount of probes, the median number of probe pairs defining a gene value is in HG-U95Av2 12 and in HG-U133A 7. With this requirement the number of genes was 7640 which is 93.7% of all common genes. Further, the overlap can be required to have selected length, which decreases the number of the usable probes and genes. Also one can select a limit for a number of probe pairs that are needed for expression value calculation for a gene. The higher this limit is the smaller is the number of genes (Figure 2).

In the case study we compared the values between technically replicated samples. We have utilized an experiment series from St Jude University [8], [9], where 132 samples of leukemia were hybridized on both array generations HG-U133A and HG-U95Av2. The correlation between samples improved significantly (Table 3). The median value of correlations increased from 0.65 to 0.74 when calculated in logarithmic scale. We determined the significance with one-way ANOVA using null hypothesis that the mean values of the distributions of correlations between array generations are the same. The F-statistic for the correlations is 168.9, and p-value is 0. Therefore, the null hypothesis can be rejected and the correlations with EGAGO are significantly better than with MAS.

Table 3. Correlations between logarithmic values of 132 technical replicates of leukemia samples.

<table>
<thead>
<tr>
<th></th>
<th>MAS5</th>
<th>EGAGO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Median</td>
<td>0.6543</td>
<td>0.7410</td>
</tr>
<tr>
<td>Std</td>
<td>0.0537</td>
<td>0.0523</td>
</tr>
<tr>
<td>Min</td>
<td>0.4300</td>
<td>0.5249</td>
</tr>
<tr>
<td>Max</td>
<td>0.7531</td>
<td>0.8248</td>
</tr>
</tbody>
</table>

The correlation values between technical replicates increases if longer overlap in nucleotides is required. At the same time, the number of genes drops dramatically. Figure 2 illustrates an example of correlation between technical replicates and number of genes when the required overlap is from 1 to 25. In addition, the number of probes that one gene is required to have can be selected. In our study, we found out that already one nucleotide long overlap increased the correlation significantly compared to MAS. The length of overlap did not have a big influence on correlations. However, if the required overlap length was greater than 15 the number of genes dropped a lot. Also, the gene values are more reliable if the values are formed based on more than only one probe. For example with overlap 25 and at least 4 probes in a gene only 254 genes will have a value. Therefore, the requirement for total overlap between probes is too tight. In our case we find it best to threshold the number of probes in each gene to be 4 and the required overlap to be 1.

We tested the comparability of technical replicates of data from all these three methods using ten samples of AML leukemia (Figure 3). In MAS5 and overlapped based masking (OL) method the gene values were set to be the median of the probe set values within each gene. The gene based redefinition of probe sets (ENSG) and EGAGO method used directly the gene values. We
tested the distributions with ANOVA and performed a multiple comparison using Tukey's honestly significant difference criterion. With significance level 0.01 we found out that the correlations of EGAGO method differ from values from other methods. Therefore the values from EGAGO analysis are more comparable than values from the other methods.

4. CONCLUSION

We have introduced a method for improving comparability between different Affymetrix array generations and compared its performance with existing methods. Our results indicate that the proposed method, EGAGO, facilitates the analysis and combined analysis of data from different generations. This should prove very useful for the construction of large-scale databases with thousands of microarray experiments from different array generations. The correlation between technical replicates increased significantly when the probe sets were formed based on the overlap between different array generations. However, if this kind of gene value calculation is used, the number of rejected probes is often very large. Therefore it might be good to consider how much data are we allowing removing, if we are aiming for an optimal result. On the other hand, how much we can trust for the information produced based on only few probes. These questions are to be answered in the future work.

5. ACKNOWLEDGMENTS

This work was supported by the Academy of Finland, the Emil Aaltonen foundation, the Foundation of Technology, the Finnish Konkordia Fund and the foundation for Commercial and Technical Sciences.

6. REFERENCES

EEG PREPROCESSING BY SOURCE DERIVATION FOR USER DEFINED EEG MONTAGE AND REFERENCE CONVERSION OF STANDARD RECORDINGS AND EVOKED POTENTIALS

Cristin Bigan¹ and Mircea Besleaga¹²

¹Ecological University of Bucharest, Romania, Biopattern partner 24,  
²Romanian Society for Clinical Neurophysiology,  
cbigan@yahoo.com

ABSTRACT

In many situations the monopolar and bipolar EEG references are not useful and the source derivation appears to be the most suitable.

Available software for EEG processing and analysis provides routines to perform software off-line reconstruction of any user defined montage from the recorded one mainly by subtraction operations performed on the original multi-channel EEG data. This montage conversion from the recorded to the computed one induces inherent errors due to the head model choice and the measured distance between electrodes variability. To improve EEG data conversion quality we propose the use of the Intrinsic Hjorth EEG preprocessing, with the advantage of replacing the variable distance between electrodes or electrodes and reference with electrical distance.

1. INTRODUCTION TO REFERENCE AND MONTAGES IN EEG

Because different locations on the head may show different electrical activities, the derivation of one particular target site to different reference sites can result in very different EEG findings. Therefore, the choice of a particular reference site may have a substantial effect on the EEG findings, which constitutes the EEG reference problem. Extensive training is required to learn the spatial interpretation of traditional EEG montages: referential, bipolar, ears, and average reference. Each EEG recording reference has its own set of advantages and disadvantages.

Traditional montages are constructed by subtracting the signal of one channel from another. Except for the average reference and the Hjorth source derivation, traditional montages give equal weights to the so-called active electrode and to the reference electrode.

2. CONVERSION PROBLEM EXPLANATORY NOTES

Various commercially available software for EEG processing and analysis provides routines to perform software off-line reconstruction of any user defined montage from the recorded one mainly by subtraction operations performed on the original multi-channel EEG data.

By example, if one ear (A1) is used as a reference and recorded as a separate channel, a linked ear reference for say Cz, a scalp electrode in the 10–20 system, can be created algebraically, \((Cz - A1) - (A2 - A1)/2 = Cz - (A1 + A2)/2\).

However, this montage conversion from the recorded to the computed one induces inherent errors due to two main factors. First the head model choice (if any) does not correspond entirely to the specific subject as a matter of spheres conductance, shape and size. Secondly, the measured distance between electrodes and from the electrodes to the used references area, varies from one subject to another.

According to [1] Hjorth was among the first to apply subtraction of signals from adjacent EEG electrodes to obtain an estimate of the surface Laplacian, noting that the Laplacian operation provides a measure of source activity as it would appear at the scalp surface. In addition to these techniques, the resolution of the EEG has been improved by methods that simply increase the spatial sampling density and „de-blur” the volume conduction effects of the skull.

In [2] the effects of the EEG montage studies have shown that a conductive wire structure alters the electric field distribution near a wire. So the electric field distribution, with a bird cage and surface coils in four different cases: coil only, coil plus head, coil plus electrodes/leads, and coil plus head and electrodes/leads, becomes less uniform and decreases in presence of the head, even for different montages of EEG electrodes, i.e., different numbers of electrodes, that were used in the simulations and even the positions of the electrodes and leads were digitized directly on the head of the subject with a digitizer.

The presentation [3] concludes there is no perfect reference for all cases. As a general principle, a known local source should be referred to an electrode distant from it. Traditional references for human Event Related Potentials have been linked mastoids, linked ears, or the nose; unfortunately none of these is unaffected by brain sources.
Errors in off-line montage reconstruction were also confirmed by direct measurement made by authors of the same subject’s EEG with two machines simultaneously, each one implementing a different reference montage and on-line recording EEG. Differences were shown after performing off-line software conversion of one montage to another, and those are explained due to the reasons shown above.

![Image](57x573 to 300x618)

Figure 1. The waveforms of the same EEG signal acquired by: (a) the recording with linked ears A1-A2 reference. (b) referenceless method of Laplacian Hjorth.

3. ELECTRICAL DISTANCE BASED SOURCE DERIVATION METHOD AND DISCUSSION

Even that according to [4] the question of the correct reference electrode has been intensively debated in the literature and the ‘reference-problem’ has been taken as a major disadvantage of EEG versus MEG because the reference indeed heavily influences waveform analyses, according to our knowledge, no major studies were taking into account the error induced by off-line conversion from one montage to another.

In order to improve EEG data quality and to reduce such errors, as an extension of [5] we propose the use of the Intrinsic Hjorth EEG preprocessing, that has the advantage of replacing the variable distance between electrodes or electrodes and reference (due to various head sizes this distance varies from one subject to another) with electrical distance defined as shown below.

As in [6] we used intrinsic version - a variation of the Hjorth Laplacian in which spatial distance is replaced by a nonspatial “electrical distance” measure reflecting the electrical similarity of electrodes. For any pair of electrodes \( i \) and \( j \), a potential difference waveform \( P_{i,j}(t) \) may be computed as:

\[
P_{i,j}(t) = P_i(t) - P_j(t)
\]

The “electrical distance” measure \( D_{i,j} \) is defined as the temporal variance of the difference potential waveform:

\[
D_{i,j} = \frac{1}{T} \sum_{t=1}^{T} (P_{i,j}(t) - \bar{P}_{i,j}(t))^2
\]

where \( T \) indicates the selected time for the montage conversion.

4. CONCLUSION

Following the goal that EEG acquisition becomes truer reflection of the electrical activity at that particular point, the source derivation method has the intention of improving localization of focal activity from the scalp for EEG and EP’s. The method enhances the activity that is unique to a particular electrode. Source derivation use in clinical EEG [5] indicated focal abnormalities more often pronounced with this technique.

As the electrical distance is based on scalp potentials differences at various time moments for each subject specific and is not fixed as a standard distance between electrodes on a a-priori model, proposed preprocessing technique directly solves also the problem of head model choice. The use of source derivation as a preprocessing step in any user defined conversion montage by off-line software reconstruction, provided a more accurate way of data visualization, closer to the recorded version of the EEG by a machine configured to record with the desired montage, as proven by tests with two ways (different montages on different machines) simultaneously recording of the same subject. The ideal physical acquisition solution for recording EEG data, closer to the source, would be by means of surface concentric electrodes, similar to the techniques used in EMG.

5. ACKNOWLEDGMENTS

Authors acknowledge BIOPATTERN EU Network of Excellence EU Contract 5088032, for the support of this study.

6. REFERENCES

A MODEL-BASED APPROACH TO CAPTURE GENETIC VARIATION FOR FUTURE ASSOCIATION STUDIES

Susana Eyheramendy1, Jonathan Marchini1, Gilean McVean1, Simon Myers2 and Peter Donnelly1

1Statistics Department, University of Oxford, OX1 3TG, UK
2Broad Institute of MIT and Harvard, One Kendall Square, Bldg 300, MA 02139, USA
(eyheram,marchini,mcvean,donnelly)@stats.ox.ac.uk, smyers@broad.mit.edu

ABSTRACT

Genome-wide association studies are still constrained by the cost of genotyping. For this reason the selection of markers or tags able to capture a significant proportion of the genetic variation is an important aspect of these studies. Most tagging SNPs selection methods have been successful in capturing the genetic variation of the data from which the tags have been chosen. However, when these tags are used in an independent dataset, a significant proportion of the remaining SNPs (non-tags) are not captured and, in most cases, there is no information on which of the non-tags are captured. We propose to use a probabilistic model to predict the non-tags based on a set of tags, as a way to capture genetic variation. An important advantage of this method is that it predicts the genotype of the non-tags which we can test for association with the phenotype. Additionally, it provides an estimate of the probabilities with which the prediction of the non-tags are made which reflects the confidence of the probabilistic model. We also propose a method to select the tagging SNPs. We empirically show using HapMap data that our approach is able to capture significantly more genetic variation than methods based solely on a pairwise LD measure.

1. INTRODUCTION

Much of the variation between people in traits such as eye or hair color, size and also disease susceptibility is heritable and has a genetic basis. Most of the genetic differences between individuals are single nucleotide polymorphisms (SNPs), which are differences in chromosomes at a nucleotide base and account for approximately 90% of the human genetic variation. It has been estimated that there are about 10 million common SNPs (frequency of each allele > 5%) across the genome. These common SNPs are presumed likely to capture a substantial proportion of phenotypic variation.

One approach to identifying the SNPs responsible for particular phenotypic traits is via association studies in which the allele frequencies of different SNPs are compared in case and control samples. A difficulty that association studies encounter is that the disease susceptibility loci are unknown and there are millions of possible sites to genotype. Even though the cost of genotyping is rapidly decreasing, it is still impractical to genotype every SNP or even a large proportion of them. Fortunately, nearby SNPs are often in strong linkage disequilibrium or, in other words, are strongly correlated with each other. Therefore, it might be possible to define a subset of the SNPs which “tag” a large proportion of the remaining variants in the genome, so that the latter would give redundant information in an association study.

There are several algorithms and methods that have been developed in the last few years which try to select the best set of tagging SNPs, e.g. Johnson et al [8], Weale et al [14]. Carlson et al [2] proposed an algorithm that aims to select a set of SNPs (the set of tagging SNPs) that ensure $r^2$ values larger than a given threshold between SNPs in the tagging SNP set and those outside the set. $r^2$ is the square of the coefficient of correlation and is one of many scores that measure the level of linkage disequilibrium (LD) between two SNPs (for a comparison of many such measures see Devlin and Risch [5]). Among the various measures, $r^2$ is particularly popular since Pritchard and Przeworski [11] showed that there is a direct relationship between the power of a test for association and the $r^2$ value.

It should be noted that, $r^2$ values between a marker and the disease susceptibility locus is just one of several parameters that can determine power. For the same $r^2$ value, the power of a particular test for association that depends on a SNP can, range from very low to very high values depending on the allele frequency and the underlying disease model. Among the other parameters that influence the power of an association study are: the disease susceptibility allele frequency, the penetrance of the disease susceptibility locus and the frequency of the alleles of the markers (see e.g. Schork [13]).

A drawback of pairwise measures of LD is that they do not capture the full correlation structure of the sequence variation. For this reason, researchers are now also investigating the properties of sets of SNPs or haplotypes. For example, Clayton [3] defines haplotype diversity as a way to find the optimal choice of tagging SNPs. Li and Stephens [9] proposed a hidden markov model (HMM) to fit haplotype data that incorporates genetic factors such as recombination rates, probability of mutation and the distance between SNPs in the model.

Much effort has been put into methods that reduce
the number of tagging SNPs required to capture genomic variation. For instance, de Bakker et al [4] propose a haplotype-based tagging method that requires significantly fewer SNPs than the algorithm of Carlson et al [2], while achieving the same coverage in the training set used to define those tags. Despite the improvement obtained with these approaches a more appropriate aim would be to find a method which will give the most efficient coverage not only for the training set but in future association studies.

In this work we propose a method to predict the non-tags as a way to capture genetic variation. Additionally, we propose an approach to select tagging SNPs that provides a list of sorted SNPs from which to choose the tags. The new methods are able to capture more of the genetic variation in a new dataset than Carlson’s algorithm given the same number of tags, as measured by three criteria (see §3). The algorithms that we propose uses the PAC (product of approximate conditionals) likelihood of Li and Stephens [9].

The paper is organized as follows. We describe the PAC likelihood in §2.1 together with a method to predict non-tags. In §2.2 we describe our novel algorithms and we discuss our results and a critical assessment of the proposed method in §3. We conclude in §4 with a discussion of our results.

2. METHODS

2.1. PAC likelihood

Li and Stephens [9] proposed a hidden markov model for haplotype data that incorporates the distance between SNPs, recombination and mutation. We use this model and the forward algorithm for hidden markov models to compute the likelihood of a set of haplotypes. More precisely, as evaluated at $S$ biallelic loci, so that $h_{1j}$ corresponds to the $i$th haplotype evaluated at the $j$th SNP. Also, assume that the next haplotype $h_{n+1}$ has some missing components, so that $h_{n+1}=(h_{n+1}^{obs},h_{n+1}^{miss})$. $h_{n+1}^{obs}$ is the vector of missing components and $h_{n+1}^{obs}$ is the vector of observed components. We use the forward algorithm to compute at each missing component $j_m$ the probability of each allele (denoted by 0 and 1) given the haplotypes $h_1,...,h_n$, the observed components $h_{n+1}^{obs}$ and a given a vector of recombination rates $p$ (which are estimated following the approach in McVean et al [10]). Thus, we compute $p_{n+1,j_m}^\star = P_R(h_{n+1,j_m}^{miss} = 1| h_1,...,h_n,h_{n+1}^{obs},p)$. We infer the missing components using the following rule:

$$l_{n+1,j_m}^{miss} = \begin{cases} 1 & \text{if } p^\star_{n+1,j_m} \geq 0.5 \\ 0 & \text{if } p^\star_{n+1,j_m} < 0.5 \end{cases}$$

(1)

In the context of the tagging SNP selection problem, the missing components correspond to the non-tagging SNPs, the observed components to tagging SNPs and $h_1,...,h_n$ correspond to the haplotypes in the training set. The reule given by equation 1 is denoted hereon as (L+S). In what follows we describe a novel algorithm to choose tagging SNPs.

2.2. Tagging SNP Selection

First, compute the difference between the PAC likelihood that excludes one SNP from each haplotype minus the PAC likelihood computed including all SNPs in the haplotypes. If haplotypes are formed with $S$ SNPs, then there are $S$ of such differences. Then, identify the single SNP that attains the minimum difference. This SNP becomes the first tagging SNP identified. This process is iterated, checking each time that the SNP to be selected is not exceeding an $r^2$ threshold with any of the SNPs already selected as tagging SNPs. The idea behind this approach is to identify the SNPs that add little or no information to the full likelihood and which in general will be SNPs that find many SNPs correlated with itself in the set of all SNPs. Specifically, the first tag selected is

$$s_1^\star = \arg\min_j l(h_1(-j),...,h_n(-j)|\rho) - l(h_1,...,h_n|\rho).$$

The second tagging SNP selected will be among the set of SNPs that are not correlated with $s_1^\star$ and that satisfy

$$s_2^\star = \arg\min_j l(h_1(-j,s_1^\star),...,h_n(-j,s_1^\star)|\rho) - l(h_1,...,h_n|\rho),$$

and so on for the following tagging SNPs to be selected. A subset of all the SNPs is returned as a list of sorted SNPs. The remaining SNPs (the ones not in the list of the returned sorted SNPs) are by construction in high $r^2$ with at least one SNP from the list.

In what follows we present the results of comparing this approach (that we denote tagLS) with Carlson’s approach for selecting tagging SNPs.

3. RESULTS

It has been suggested that the populations genotyped in the HapMap project [7] may serve as reference populations for the selection of tagging SNPs in association studies. In addition to surveying variation genome wide, the HapMap Project focused on 10 ENCODE regions for comprehensive genotyping as part of an in-depth study of human genetic variation.

We measure the performance of our approach and compare it with Carlson’s approach in the 10 ENCODE regions, which are each roughly 500 kb in length. Individuals from three populations were genotyped: 60 unrelated Europeans from Utah (CEU), 60 unrelated Africans from Nigeria (YRI) and 89 unrelated Asians (Han Chinese (HCB) and Japanese from Tokio (JPT)). We consider only SNPs with MAF bigger than 5% within each population.

The comparison of the two algorithms tries to assess the performance of a set of tagging SNPs in a future association study by randomly assigning haplotypes from each of the three populations into equally sized training and test datasets. The training dataset is used to perform the tagging SNP selection while the test dataset is not used to define the tagging SNPs and thus is a proxy for the the genotypes obtained in the future study.
We measure the performance of the algorithms using three criteria: proportion of non-tagging SNPs “captured”, misclassification rate and the Brier score (Brier [1]). Henceforth, we say that a non-tagging SNP is “captured” if the prediction of the non-tag has \( r^2 > 0.8 \) with the actual value of the non-tag in the test dataset. Recall that by using the rule given by Equation 1 we can predict every non-tagging SNP. Therefore, for each non-tagging SNP, there is a predicted SNP that represent our “best” guess for that SNP. In a similar way, the “best” guess for each non-tag in the test data that Carlson’s algorithm provides is the tag in the test data with which the non-tag has the highest \( r^2 \) value. The proportion of SNPs captured is given by the sum of the number of tagging SNPs and the number of non-tags captured divided by the total number of SNPs. Misclassification, for each SNP, is computed as the number of mismatches between the true SNP and its predicted SNP (or its negative depending on which gives smaller misclassification). The overall misclassification rate is given by the sum of the misclassification given by each non-tag divided by the total number of SNP and the total number of haplotypes in the test data. If the rule by given 1 is used to make the predictions, the Brier score is computed in the following way:

\[
\sum_{i=n+1}^{2n} \frac{\left(p_{i,j}^* - h_{ij}\right)^2}{2},
\]

where \( NT \) correspond to the set of non-tagging SNPs and the sum is over the haplotypes in test data.

If instead we use Carlson’s algorithm to predict the non-tags, then we replace the probability \( p_{i,j}^* \) in the above expression by either a 1 or a 0 depending on the value of the tag, evaluated at haplotype \( j \), that has the highest \( r^2 \) value with the non-tag that we are trying to predict.

3.1. Generalization performance

It is well recognized that point estimators of \( r^2 \) have a high sampling variance (e.g. Ewens [6]) and therefore might indicate that one SNP captures another when in fact it does not. Carlson et al [2] used simulated data to empirically test the reliability of the proportion of SNPs captured using \( r^2 \) values and different thresholds and concluded that thresholds > 0.5 appear to yield more reliable results for the particular sample size that they used. We use Carlson’s algorithm on the training set to choose the tags, using a cutoff of 0.8 and then assess its performance in capturing variation in the test datasets.

We stopped Carlson’s algorithm after 50 and 100 SNPs were selected as tagging SNPs and measured the percentage of SNPs captured by these sets of SNPs and compared it with what is expected from training data (Table 1). For example, with 50 tags, Carlson’s tags capture 84% of the total number of SNPs in the combined Asian populations in trainin data (‘tr’ in Table 1). When the same SNPs are considered in test data (‘te’ in Table 1), 73% of the SNPs are captured. We note that regardless of the number of tagging SNPs chosen, when considering a new dataset approximately 12% of the SNPs captured in training data are not captured in a new dataset. Note also that the African population (YRI) requires more SNPs to capture the same proportion of non-tags than the European (CEU) or combined Asian populations (JPT+HCB). This agrees well with the evidence for slightly higher genetic diversity in the African populations, a fact that has been taken as evidence for the “out of Africa” model (see e.g. Reich et al [12]).

Table 1 shows misclassification rate of Carlson’s tagging SNPs in training and test data, averaged over the 10 ENCODE regions and 10 training-test splits of the data for 50 and 100 tags. In the three populations, about 1 – 3% of increase in the misclassification rate occurs when we test the tagging SNPs in an independent dataset.

3.2. Carlson’s prediction vs L+S prediction

To assess whether we could gain information from a probabilistic model in capturing non-tagging SNPs, we use the same set of tagging SNPs that Carlson’s algorithm finds to predict all SNPs outside the tagging SNP set using the Li and Stephens likelihood as explained in § 2.1. In this case the tagging SNPs correspond to the observed components of the haplotype and the remaining SNPs corresponds to the missing components. The missing components are predicted using the rule given by equation (1). We estimate the \( r^2 \) value between all SNPs outside the tagging SNP set with their predicted values using test data. The difference in the percentage of SNPs captured by Carlson’s algorithm compared to the ones captured by predicting the non-tags using the Li and Stephens model is more noticeable when the number of tagging SNPs is smaller. Using the Li and Stephens model to predict non-tagging SNPs allows an increase of 11% in the captured SNPs in the YRI population when 50 and 100 SNPs are in the tagging SNP set and between 6 – 8% in the JPT+HCB and CEU populations for the same number of SNPs (Table 1).
then the gain obtained by predicting SNPs based on the Li and Stephens [9] model can be important.

Another way to assess the performance of the tagging SNPs is misclassification rate, which reflects the overall error in prediction. The results for Carlson’s approach and our approach in training and test data are shown in Table 2. The misclassification rate of Carlson’s tagging SNPs decreases slightly in test data compared with the performance in training data and again we can see that the gain is about 2–4% when we predict the non-tags using L+S.

We have described a way in which we can capture more of the genetic variation by using the tagging SNPs chosen by Carlson’s algorithm and predicting the non-tagging SNPs using the Li and Stephens [9] model. The next section shows that if we additionally use the Li and Stephens [9] model to choose the tags as described in §2.2 we are able to capture even more of the genetic variation.

### 3.3. Performance of Carlson’s tags vs tagLS

We now assess the performance of the tags tagLS. This set of tags is used to predict the non-tags using the LS model and the rule given by Equation 1. We use 10 training-test splits of the data to provide error estimates in the evaluation of performance.

The results of the Brier scores, misclassification and proportion of SNPs captured, averaged over the 10 ENCODE regions and the 10 training-test splits, are shown in Table 3. Table 1 shows a slight improvement of 1% – 3% in the proportion of SNPs captured by tagLS compared to the performance when the tags have been chosen using Carlson’s algorithm. Misclassification remains roughly the same for the two sets of tags. Brier score is slightly smaller in the three population when tagLS are used to predict the non-tags, which reflects better confidence of the prediction model (see Table 3).

### 4. DISCUSSION

The ability to predict the non-tags offers several advantages. The most obvious one is that it provides extra SNPs, besides the tagging SNPs, with which we can test for association with the phenotype. Additionally, it provides an estimate of the probabilities with which the prediction of the non-tags are made which reflects the confidence of the probabilistic model.

Our methods were developed favoring ease of computation and fast implementation rather than exact calculations. Therefore, there are still several directions that could lead to improvement of the methods proposed. One possibility is to optimize the algorithm that searches for the best set of tags. Another possibility is to do the prediction of the non-tags taking into account the dependence between the SNPs, which we are not considering in this work. These possibilities will be explored in future work.

### Acknowledgements

S. E. would like to thank Daniel Falush for many discussions and useful comments that improved the presentation of this work and Korbinian Strimmer for useful comments who part of this work was done.

### 5. REFERENCES


---

Table 2. Misclassification rate averaged over the 10 ENCODE regions and 10 training-test splits.

<table>
<thead>
<tr>
<th>Population</th>
<th>Data</th>
<th>tag</th>
<th>prediction method</th>
<th>num tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>CEU</td>
<td>tr</td>
<td>Carlson</td>
<td>Carlson</td>
<td>0.04</td>
</tr>
<tr>
<td>JPT+HCB</td>
<td>tr</td>
<td>Carlson</td>
<td>Carlson</td>
<td>0.03</td>
</tr>
<tr>
<td>YRI</td>
<td>tr</td>
<td>Carlson</td>
<td>Carlson</td>
<td>0.07</td>
</tr>
<tr>
<td>CEU</td>
<td>te</td>
<td>Carlson</td>
<td>Carlson</td>
<td>0.04</td>
</tr>
<tr>
<td>JPT+HCB</td>
<td>te</td>
<td>Carlson</td>
<td>Carlson</td>
<td>0.04</td>
</tr>
<tr>
<td>YRI</td>
<td>te</td>
<td>Carlson</td>
<td>Carlson</td>
<td>0.08</td>
</tr>
<tr>
<td>CEU</td>
<td>te</td>
<td>tagLS</td>
<td>L+S</td>
<td>0.02</td>
</tr>
<tr>
<td>JPT+HCB</td>
<td>te</td>
<td>tagLS</td>
<td>L+S</td>
<td>0.02</td>
</tr>
<tr>
<td>YRI</td>
<td>te</td>
<td>tagLS</td>
<td>L+S</td>
<td>0.04</td>
</tr>
</tbody>
</table>

Table 3. Brier score when tagLS and Carlson’s tags (resp.) are used and the non-tags are predicted using rule given by equation 1 (L+S).
STATISTICAL CONSERVATION ANALYSIS OF ZINC-INTERACTING RESIDUES

Ioannis N. Kasampalidis, Ioannis Pitas and Kleoniki Lyroudia

1Department of Informatics, Aristotle University of Thessaloniki, 2Department of Endodontology, School of Dentistry, Aristotle University of Thessaloniki, Thessaloniki, 54124, Greece
pitas@aiia.csd.auth.gr

ABSTRACT

As a result of rapid advances in genome sequencing, the pace of discovery of new protein sequences has surpassed that of structure/function determination by orders of magnitude. This is also true for metal-binding proteins, i.e. proteins that bind one or more metal atoms necessary for their biological function. With regard to metal-interacting residues, the question arising is whether these interactions apply additional evolutionary constraints and to what extent. We try to answer this question for a subset of metal-binding-proteins, namely zinc-binding proteins, which play an important role in a number of biological processes, as exemplified by the tumor-suppressor protein p53. Our results show significantly higher evolutionary pressure on zinc-interacting residues, a result which can be used in a number of other studies, including zinc-binding site prediction.

1. INTRODUCTION

Bioinorganic chemistry is the field dealing with the crucial interactions between inorganic metals and biological molecules of interest [1]. An important subset of biological molecules, metallo-proteins, plays a fundamental role in numerous biological processes, as evidenced by the fact that about one third of determined protein structures contain metal-binding sites, as shown by a simple Protein Data Bank (PDB) search [2].

A great deal of effort has recently been devoted to the analysis and prediction of metal-binding-sites. Many researchers have focused on the analysis of their structure, either with respect to its geometry as in Harding et al. [3] and Tainer et al. [4] or its chemical properties as in Karlin et al. [5]. Other studies perform analysis based on density functional theory/continuum dielectric methods as in Dudev et al. [6]. Such analysis can have significant impact on better functional characterization of metal-binding proteins, drug design, database searching for metal-binding-proteins, as in Andreini et al. [7], or metal-binding-site (MBS) prediction.

Regarding MBS prediction, a number of recent approaches have been proposed, including an energy-based method by Laurie et al. [8], a support-vector machine predictor of cysteine binding state by Passerini and Frasconi [9], and a recursive neural network predictor of disulfide bridge connectivity by Vullo and Frasconi [10].

A key piece of information in some of these methods [9, 10] is conservation. Their results are encouraging; however, no large-scale analysis has been performed on conservation of metal-interacting residues. Such a study could provide justification for using conservation information as a feature for MBS prediction. Additionally, it could provide further insight to the functional importance of certain metal/residue combinations by comparison of the extent of residue conservation among different metal ions and protein families.

In this study, we focus on conservation analysis of zinc-interacting residues. Zinc is one of the metals playing a crucial role in a number of biologically significant proteins, including p53 [11], a tumor-suppressor protein. This work investigates whether there is higher selective pressure on zinc-interacting residues vs. non-zinc-interacting residues.

For this purpose, we derived a non-redundant set of zinc-interacting proteins from PDB. Because this set did not contain enough members for a large-scale analysis, we used homology search via PSI-BLAST [12] to obtain additional putative zinc-interacting proteins, limiting our selection to one ortholog protein from each species. The known-structure proteins were grouped according to families as defined in Structural Classification of Proteins (SCOP) [13], while their orthologs were also included in the same family. Protein grouping by family was preferred since same family membership in SCOP indicates clear evolutionary relationship. A multiple sequence alignment (MSA) was performed on all members of each family.

We applied two approaches in our analysis. In the first approach, we measured the identity ratio, in the MSA, for all zinc-interacting residues and compared it to the identity ratio for non-interacting residues. This approach was limited only to known-structure sequences, since for unknown-structure sequences the metal-interacting residues cannot be guaranteed. The mean identity ratio of all residues within a family was calculated and the means of all families were compared.
We also pursued conservation analysis based on an information theoretic approach, where we calculate separate substitution matrices for zinc-interacting and non-zinc-interacting MSA columns for each family. These matrices were compared using the relative entropy metric, as described in Altschul [14]. This metric serves as a distance measure between the actual and the theoretically expected probability distribution of residue substitutions, where higher relative entropy indicates higher selective pressure.

2. METHODS

2.1. Dataset

We created a dataset of zinc-interacting structures with the help of PDB [2] using an appropriate query, where we required structures to have resolution better than 2.5Å and no mutant residues. From the structures meeting these criteria, we chose only the ones classified in the following SCOP classes: 1) all alpha proteins, 2) all beta proteins, 3) alpha and beta proteins (a+b), 4) alpha and beta proteins (a/b) and 5) membrane and cell surface proteins and peptides. We also required sequences to have a length greater than 40 residues. A non-redundant set was derived from the proteins meeting these criteria with the help of the algorithm by Li et al. [15], at the 90% identity level, as implemented in PDB. In total, 481 PDB files containing zinc were identified, where some of these files may belong to more than one class. The alpha class contained 105 files, the beta class 165, the a+b class 177, the a/b class 174 and the class of membrane and cell surface proteins 1 PDB file.

2.2. Zinc-interacting residues

For each structure, the metal-interacting residues were identified using a distance cut-off of 4Å from the metal atom. Although this criterion does not take into account the biological significance of this interaction, it is probably the best criterion currently available for automated metal-interacting residue identification. The distance cut-off of 4Å was chosen as an upper empirical bound, as described in Harding et al. [3]. Only the domains, as defined in SCOP, containing zinc-interacting residues were selected for multiple sequence alignment and these domains were afterwards grouped by SCOP family. A small number of domains from the original set were discarded because their species could not be identified based on the NCBI taxonomy database [16, 17].

2.3. Multiple sequence alignments(MSA)

Initial multiple sequence alignments for the domains containing metal-interacting residues were performed using PSI-BLAST (2) against the NCBI NR database [16, 17], with an e-value cutoff of $10^{-3}$. In order to identify orthologs, for each protein, we selected only the reciprocal best hit from each species in the PSI-BLAST reports. In the reports, sequences corresponding to the same species as the query sequence were also discarded.

Sequences were further filtered by discarding entries with the following keywords: synthetic, putative, probable, predicted, hypothetical, unnamed, unknown, unidentified, designed, vector. The resulting sequences were grouped with known-structure sequences into SCOP families and the multiple sequence alignments were further refined using MUSCLE (multiple sequence comparison by log-expectation) [18].

2.4. Identity ratio

The identity ratio for a single residue was calculated as the ratio of identical residues and the length of the MSA column. This ratio was calculated only for residues of known-structure sequences, while sequence gaps in the MSA columns were not included in the calculation. The mean zinc-interacting and non-interacting identity ratios of each family were calculated by simple averaging over all zinc-interacting and non-interacting residues of each family respectively.

2.5. Substitution matrices

Within each family MSA, a MSA column was defined as zinc-interacting if it contained at least one zinc interacting residue. Substitution matrices were created separately for non-zinc-interacting and zinc-interacting MSA columns, using all the sequences in each MSA, as described in Henikoff & Henikoff [19]. More specifically, each element $s_{i,j}$ of the substitution matrix is calculated as in

$$s_{i,j} = \log_2 \left( \frac{c_{i,j}}{e_{i,j}} \right), \quad (1)$$

where $c_{i,j}$ and $e_{i,j}$ are the observed and expected frequencies respectively. The observed frequencies are calculated separately for the zinc-interacting and non-interacting MSA columns. The expected frequencies are calculated from all MSA columns using the formula described in Henikoff and Henikoff [19]. The matrices were then compared based on the relative entropy metric, shown in (2), as described in Altschul [14].

$$H = \sum_{i,j} e_{i,j} \times s_{i,j}, \quad (2)$$

where $c_{i,j}$ and $s_{i,j}$ are the observed frequency and the elements of the substitution matrix respectively.

3. RESULTS

3.1. Identity ratio

The mean identity ratio for zinc interacting residues is 0.7, while for non-interacting residues, it is 0.51. The t-test on the two means resulted in a p-value of $7.25 \times 10^{-23}$, which is highly significant. The histogram for the mean identity ratio per family is shown in Figure 1. This result clearly shows the higher evolutionary constraints for zinc-interacting residues of known-structure sequences.
3.2. Relative Entropy

Identity ratio analysis on known-structure sequences does not provide a full picture of each family MSA, since it focuses only on sequences of known-structure. For this reason, substitution matrices were created for each family, for zinc-interacting and non-zinc interacting columns. Relative entropy was calculated for each of these matrices and the histogram for all 212 families is shown in Figure 2. The mean relative entropy for zinc interacting MSAs is 6.35, while for non-interacting residues it is 1.42. The difference of the two means is highly significant, as indicated by the t-test p-value of $1.84 \times 10^{-53}$.

4. CONCLUSION

In this study, we pursued analysis of zinc-interacting proteins' conservation. Zinc-interacting proteins take part in a number of important biological process as exemplified by the tumor-suppressor protein p53. Our statistical methodology showed significantly higher conservation of zinc-interacting residues compared to non-zinc-interacting residues. This conclusion is drawn from two types of metrics, identity ratio, which is based only on known-structure sequences, and relative entropy, which is based on all orthologous sequences.

However, a great deal of work remains to be done. More specifically, analysis needs to be extended to other biologically significant metal ions. Moreover, the conservation levels between different families need to be compared, in order to extract useful biological hindsight into metal-binding site structure and function. The completion of these studies can have significant implications for metal-binding site prediction, protein functional characterization and drug design.

5. ACKNOWLEDGMENTS

This work was supported by the EU project Biopattern: Computational Intelligence for biopattern analysis in Support of eHealthcare, Network of Excellence Project No. 508803.

6. REFERENCES


BAYESIAN ORTHOGONAL LEAST SQUARES (BOLS) ALGORITHM FOR REVERSE-ENGINEERING GENE REGULATORY NETWORKS

Chang Sik Kim\(^1\), Tapio Salakoski\(^2\) and Mauno Vihinen\(^1\)

\(^1\)Bioinformatics Group, Institute of Medical Technology, FI-33014 Tampere, Finland
\(^2\)Department of Information Technology, University of Turku, FI-20520 Turku, Finland

sik.kim.chang@uta.fi, mauno.vihinen@uta.fi, and tapio.salakoski@it.utu.fi

ABSTRACT

We present an efficient algorithm for reverse engineering gene regulatory networks from microarray datasets using linear system of ordinary differential equations dealing with underdetermined and ill-conditioned issues. Our method was evaluated in \textit{in silico} experiments. It was shown that the method can be readily applied to reconstruct the sparse network structure for a linear system with relatively small number of data points. The algorithm only requires \(O(K(2N-1))\) computational complexity, where \(K\) is the number of genes and \(N\) is the number of data points in the dataset. The algorithm can be also used to reconstruct partial network structure with extremely small number of data points. The method was successfully applied to predict networks and to interpret yeast cell cycle gene expression data.

1. INTRODUCTION

Development of efficient computational methods to elucidate gene regulatory networks is one of great challenges in systems biology. Networks with large numbers of genes/proteins/components will likely require stronger optimization algorithms. It has already been proven that linear systems of ordinary differential equations are useful for modeling simple gene regulatory systems [1]. We study a simple system of ordinary differential equations using continuous variables and propose an efficient reverse engineering algorithm for underdetermined and ill-conditioned linear model, Bayesian Orthogonal Least Squares (BOLS) algorithm. Reverse engineering with linear model using small gene expression dataset is mathematically challenging because of two reasons. Firstly, the system is underdetermined, which means that there is extremely large number of genes (\(K\)) than data points (\(N\)), i.e., \(N<<K\). The number of data points e.g. in microarray experiments is limited due to the cost of experiments. Secondly, a system of linear model is ill-conditioned because the solution is very sensitive to the peculiarities on data (such as noise). Three techniques were studied and combined to overcome the two obstacles and to create the BOLS algorithm: 1) orthogonal least square method (OLS) [2], 2) second order derivative for network pruning [3] and 3) Bayesian model comparison [4].

2. METHOD

The linear system of ordinary differential equations was used for the modeling of a gene regulatory network. We first study three techniques and combine them to create the BOLS algorithm. A system of a genetic regulation using linear differential equations is described:

\[
\frac{d e_i(t)}{d t} = \sum_{j=1}^{K} w_{ij} e_j(t) + \varepsilon_i(t) \quad \text{for } i = 1, 2, \ldots, K
\]

(1)

Here, \(e_i(t)\) is the level of mRNAs at any given time \(t\), which influence the expression levels of the genes, \(\varepsilon_i\) represents noise, and \(\Delta t\) represents the time interval between expression measurements. The value \(w_{ij}\) describes the strength of influence of the \(j\)th gene on the \(i\)th gene during time interval \(\Delta t\). The positive, negative, and zero values of \(w_{ij}\) indicate activation, inhibition, and no interaction, respectively. We can use a linear system for reverse engineering if Eq. 1 is rewritten as

\[
Y_i = E w_i
\]

(2)

where \(i = 1, 2, \ldots, K\) and \(K\) is the number of genes in the dataset. \(Y_i\) and \(w_i\) are column matrices and are defined as

\[
Y_i = \{de_i(1)/dt, de_i(2)/dt, \ldots, de_i(N)/dt\}^T \quad \text{and} \quad E = \{e_1, e_2, \ldots, e_K\}^T
\]

The OLS [2] method involves decomposition of the design matrix into two by Gram-Schmidt Orthogonalization theory as

\[
E = XU
\]

where \(E = \{e_1, e_2, \ldots, e_K\}\), \(X = \{x_1, x_2, \ldots, x_K\}\), and \(U\) is a \(K\) by \(K\) triangular matrix with 1’s on the diagonal and 0’s below the diagonal. Let’s say that \(w\) is the regression parameter inferred by \(E\) and \(g\) is the regression parameter inferred by \(X\). It is noted that \(g\) and \(w\) satisfy the triangular system.
Figure 1. The evaluation of performance of the BOLS algorithm as the function of numbers of data points on simulated datasets. The noise level was set to approximately 0.072, and the complexity of the network \( m_{\text{max}} \) was 4. (100 (□), 300 (O), 500 (∆), and 800 (∇) genes)

\[ g = Uw. \]

Because \( x_i \) and \( x_j \) (\( i \neq j \)) are orthogonal to each other, the error reduction ratio only due to \( x_i \) can be defined as:

\[ [\text{NError}]_i = (x_j^T x_j)g_j^2 / (Y^TY), \quad 1 \leq j \leq K. \]

This error term provides a simple and efficient measure for seeking a subset of significant regression parameters in a forward-regression way. We have to reduce unnecessary parameters to deal with the ill-conditioned problem. We can obtain the optimal solution by trading off between the complexity of model and the data misfit [4].

We start this procedure with a very small value for data misfit. As the complexity of the model is reduced; i.e. the number of effective parameters is reduced, the value of data misfit is increased. The optimal complexity of the model for “true solution” is decided using a Bayesian model comparison frame. With a second order derivative for network pruning [3], we can select the parameters, whose elimination produces the least increase of cost function C.

With Bayesian frame [4], we can compare alternative models when our model structures keep changing with second order derivatives for network pruning method. We have a multivariate linear model. Let’s say we have a dataset \( D = \{Y, X\} \), where \( Y = \{y_1, y_2, \ldots, y_N\}^T \) is the target dataset and \( X = \{x_1, x_2, \ldots, x_K\}^T \) is the input dataset, or so called design matrix. The regression parameters we want to infer are \( g = \{g_1, g_2, \ldots, g_K\}^T \). The log posterior probability of data D given \( \alpha \) and \( \beta \) can be derived [4] as:

\[ \log P(D | \alpha, \beta, H_i) = -\frac{1}{2} \beta (Y - Xg)^T (Y - Xg) - \frac{1}{2} \beta g^T g - \frac{1}{2} \log |A| - \frac{k}{2} \log(e) - \frac{N}{2} \log(\beta) - \frac{N}{2} \log(2\pi), \]

where the subscript MP denotes to Most Probable. The evidence \( P(D|H_i) \) can be obtained if we marginalize the probability defined in Eq. 4 over the hyper-parameters \( \alpha \) and \( \beta \). Before the marginalization of the probability defined in the above equation, we have to find the most probable value of the hyper-parameters \( \alpha \) and \( \beta \). The differentiation of Eq. 4 over \( \alpha \) and \( \beta \) and the rearrangement gives formulae for iterative re-estimation of \( \alpha \) and \( \beta \) [4].
\[ \hat{\alpha} := \frac{\gamma}{g' \cdot g} \quad (5) \]
\[ \hat{\beta} := \frac{K - \gamma}{(Y - \lambda g)' \cdot (Y - \lambda g)} \quad (6) \]

where \( \gamma = N - \alpha \text{Trace}(A^{-1}) \), \( N \) is number of data points, and \( K \) is number of variables (genes). To rank alternative structures (or complexities) of the model in the light of the dataset \( D \), we evaluate the evidence by marginalizing the posterior probability \( P(D|\alpha, \beta, H_i) \) over \( \alpha \) and \( \beta \),

\[ P(D|H_i) = \int \int P(D|\alpha, \beta, H_i) P(\alpha, \beta|\text{prior}) d\alpha d\beta \]

We have very little prior information about \( \alpha \) and \( \beta \). When the available prior information is minimal, the learning process is often started with an objective prior probability. This uninformative prior is called as “vague prior” for a parameter on \((0, \infty)\), which is a flat prior. This prior can be left out when we compare alternative models. With the prior available, we can marginalize the posterior \( P(D|\alpha, \beta, H_i) \). The marginalization of \( P(D|\alpha, \beta, H_i) \) over \( \alpha \) and \( \beta \) can be estimated using a flat prior and Gaussian integration [4],

\[ \log[P(D|\alpha, \beta, H_i)] = \log[P(D|\alpha, \beta, H_i)] \cdot \frac{\sqrt{\sigma_{\alpha \beta}}}{\sigma_{\alpha \beta}} \cdot \frac{\sqrt{\sigma_{\beta \beta}}}{\sigma_{\beta \beta}} \]
\[ \times \log[P(D|\alpha, \beta, H_i)] \cdot \frac{\sqrt{\sigma_{\alpha \beta}}}{\sigma_{\alpha \beta}} \cdot \frac{\sqrt{\sigma_{\beta \beta}}}{\sigma_{\beta \beta}} \]
\[ \left(7\right) \]

where \( \sigma_{\alpha \beta} \) and \( \sigma_{\beta \beta} \) are the error bars on log\( \alpha \) and log\( \beta \), found by differentiating Eq. 4 twice:

\[ \sigma_{\alpha \beta}^2 \approx \frac{2}{\gamma} \]
\[ \sigma_{\beta \beta}^2 \approx \frac{2}{(N - \gamma)} \]

The three studied methods were combined to a reverse engineering algorithm. We have Eq. 2 for reverse engineering with \( K \) number of genes with \( N \) number of data points. The inputs to this algorithm are a certain gene as a target gene and the other genes as possible regulator candidates. The output from the algorithm is defined as an unit-network, which consists of a target gene and a list of the most probable regulators of the target gene. The algorithm is run \( K \) times and all the output unit-networks can be combined to create the complete regulatory network.

### 2.1 Pseudo algorithm of BOLS for a unit-network construction:

Step 1: Set gene \( j \) as a target gene and set the rest of the genes as possible regulator candidates
Step 2: Train Eq. 2 using OLS
Step 3: Find the most probable hyper-parameters \( \alpha \) and \( \beta \) with the iterative re-estimation Eq. 5 and 6.
Step 4: Compute the evidence \( P(D|H_i) \) for the current state network \( H_i \) with Eq. 7.

Step 5: Find the parameter \( w_j \) that gives the smallest \( L_j \) by Eq. 3, and delete \( w_j \)
Step 6: Go to Step 3 if the number of parameters on network is greater than one.
Step 7: Select the network that has the maximum evidence \( P(D|H_i) \).

### 3. RESULTS AND DISCUSSION

We tested the method and evaluated with in silico experiments. Random sparse networks were created and the expression data were generated. To create a sparse network structure, we proceeded unit-network by unit-network. For each unit-network, a random number \( m \) was picked, which should be less than or equal to \( m_{\text{max}} \). We then randomly selected \( m \) genes and assigned each of them with random nonzero values. If \( m_{\text{max}} \ll K \), a sparse network structure could be created. The complexity of the network can be varied by changing \( m_{\text{max}} \) value. The effect of the number of genes (\( K \)), the number of generated data points (\( N \)), and complexity of network structure (\( m_{\text{max}} \)) were investigated. The performance of the algorithm was measured by three values; number of errors, sensitivity, and precision. The number of errors is the simple count of incorrectly recovered interactions. The sensitivity is defined as the percentage of correctly recovered interactions among true interactions. These “true” interactions are randomly created networks during simulation process. The precision is defined as the percentage of correctly recovered interactions among all recovered interactions. As the number of data points on simulated dataset increases with fixed number of genes, noise level, and complexity of network, the performance of the BOLS algorithm increases (Fig. 1). The number of errors decreased as the number of data points increased with fixed complexity of the networks. Therefore, more complex networks need more data points to recover the network correctly (Fig 2).

The BOLS method was applied to reverse engineer genetic networks from yeast cell cycle microarray dataset of 17 time points [5]. There are a number of hubs, which are key regulators for many genes (Fig. 3A). When visualizing the data by cell cycle stages, it becomes evident that the BOLS derived networks clearly define the subnetworks in each stage. In addition, the hubs are linked only to genes, which are expressed in the same stage or in immediately earlier or later stage. The G1 stage hub in Figure 3B is ribonuclease H, which is essential for hydrolysing RNA bound to complementary DNA. The S. cerevisiae RNase H2 consists of three proteins. The network also contains other DNA binding/processing proteins. \( \alpha 1 \) protein Hmlalpha1 (Ycl066w) is involved in the regulation of mating type specific gene expression. Together with MADS-box family DNA-binding protein Mcm1, \( \alpha 1 \) induces DNA bending assumed to be essential for
transcription activation. Cde9 has DNA ligase I activity required for DNA repair, recombination and replication. Slx4 (Ylr135w) together with Slx1 (Ybr228w) forms a complex that hydrolyzes branches in duplex DNA. The complex is active in a stalled or converging replication fork. Only one of the genes in this system, coding for serine protease Prb1 (Yel060c), is not DNA binding or processing.

We proposed an efficient reverse engineering algorithm to overcome underdetermined and ill-conditioned problems in the linear model. Previously, a method using singular value decomposition (SVD) and robust regression was developed to overcome these issues in the linear model [6]. Because SVD leads to non-unique solutions, they imposed sparseness as an additional constraint to select the “true” solution from the entire family of SVD solutions. Earlier gene regulatory network studies have indicated that gene regulatory networks in most biological systems are sparse. By combining three techniques, we present a method to produce optimal ‘sparse’ network structures motivated by information theory.

BOLS algorithm is computationally efficient, because it only requires $O(K(2N-1))$ computations. We are confident that our method can be readily applied to reconstruct the sparse network structure for a linear system with relatively small sampling complexity. In addition to microarray data, the method can be also used with other large datasets presenting function of biological molecules.

4. REFERENCES


ABSTRACT

We present a novel model of excitation-contraction coupling of mouse ventricular myocyte. The model builds on a detailed and physiologically based description of intracellular calcium dynamics that include action potential properties, ion-channel kinetics and enzyme reaction chain. The most important novel feature of the model is the rate-dependent regulation that has three targets in the Ca\(^{2+}\) handling of a cardiac myocyte. A few excitation-contraction coupling models for the mouse ventricular myocyte have been published during recent years but they all lack rate-dependent regulation. Our modeling shows that the implemented regulatory mechanisms induce sought-after changes in the targets. We also conclude that the presented activity scheme of calmodulin-dependent protein kinase II is suitable mediator for the frequency encoding.

1. INTRODUCTION

The complex process from electrical excitation of cardiac myocyte to contraction of the tissue is commonly referred to as excitation-contraction coupling (ECC). In this process of cardiac electrical activity, calcium (Ca\(^{2+}\)) acts as an essential second messenger and a direct activator of myofilament contraction [1].

Within each cycle of electrical activity, i.e., during cardiac action potential (AP), Ca\(^{2+}\) enters the cell through L-type calcium channels (LTCCs) as inward Ca\(^{2+}\) current (I\(_{\text{Ca,L}}\)) that triggers Ca\(^{2+}\) release from the sarcoplasmic reticulum (SR) via the ryanodine receptor (RyR) channel [1]. The relative contribution of these two Ca\(^{2+}\) sources varies depending on species, for mouse the SR Ca\(^{2+}\) release is about an order of magnitude larger than I\(_{\text{Ca,L}}\) [2]. In order to avoid cumulative gain or loss of cytosolic and SR Ca\(^{2+}\), there has to be a balance between Ca\(^{2+}\) influx and extrusion. The rise in the free cytosolic Ca\(^{2+}\) concentration following calcium-induced calcium release (CICR) requires removal of Ca\(^{2+}\) from cytosol for the relaxation to occur. This transport has four pathways that involve SR Ca\(^{2+}\) ATPase (SERCA), sarcoplasmic Na/Ca\(^{2+}\) exchange (NCX), sarcoplasmic Ca\(^{2+}\) ATPase and mitochondrial uniport [1]. In a mouse cardiac myocyte, the latter two mechanisms have very small contributions to Ca\(^{2+}\) removal, 0.1% and 0.5% respectively [2]. Whereas SERCA has a dominant role (90%) in rapid removal of Ca\(^{2+}\) from the cytosol, which leaves a smaller part (9%) for the NCX.

The aim of this work was to develop a new mathematical model of a mouse ventricular myocyte. The mouse is perhaps the most commonly used animal model for studies of cardiac electrophysiology. Therefore, it is a logical choice for mathematical modeling. The fact that there are nowadays a variety of transgenic and gene knockout mice available, makes it an even more interesting choice.

The calmodulin-dependent protein kinase II (CaMKII) is a holoenzyme that goes through a process of autophosphorylation, i.e., the phosphorylated state continues even after the Ca\(^{2+}\)-CaM complex has been removed, which leads to prolonged activation of CaMKII [3]. It has been shown that the enzyme can decode the frequency of Ca\(^{2+}\) spikes into distinct amounts of kinase activity [4]. This feature of CaMKII and the fact that it has many phosphorylation targets in the Ca\(^{2+}\) transport machinery has made it an interesting research subject in studies related to rate-dependent cardiac Ca\(^{2+}\)-handling.

An important motivator for the studies is the fact that CaMKII has been verified as a proarrhythmic signaling molecule and may therefore offer a novel solution for antiarrhythmic therapy [5], [6].

2. MODEL SPECIFICATION

A schematic drawing of the model is presented in Figure 1. In addition to the ionic currents and fluxes of Ca\(^{2+}\), the model includes the descriptions of other ionic balances as well, but in the figure only the ones that are most closely related to Ca\(^{2+}\) handling are presented. The descriptions of ionic currents and Ca\(^{2+}\) fluxes within the cell are based on a previously published scheme [7].

The CaMKII part of the model is based on a reaction scheme published by Bhalla [8]. This submodel is presented as a control diagram in Figure 2 and it includes the following main components: calmodulin (CaM), cal-
The Ca\textsuperscript{2+} transport mechanisms are L-type Ca\textsuperscript{2+} current (ICaL), ryanodine receptor (RyR), SR Ca\textsuperscript{2+} ATPase (SERCA), and sarcolemmal Na/Ca\textsuperscript{2+} exchange (NCX). They are regulated by calmodulin kinase II reaction chain (CaMKII*) and phospholamban (PLB). The Ca\textsuperscript{2+} fluxes within the cell are uptake of Ca\textsuperscript{2+} from the cytosol to the network sarcoplasmic reticulum (NSR) (JUP), Ca\textsuperscript{2+} release from the junctional SR (JREL), Ca\textsuperscript{2+} flux from the network SR (NSR) to junctional SR (JSR) (JTR), Ca\textsuperscript{2+} leak from the SR to the cytosol (JLEAK) and Ca\textsuperscript{2+} flux from the subspace (SS) volume to the bulk myoplasm (JXFER). The cytosolic bulk Ca\textsuperscript{2+} concentrations is [Ca\textsuperscript{2+}]\textsubscript{i}. The concentrations in the SS, JSR and NSR compartments are [Ca\textsuperscript{2+}]\textsubscript{SS}, [Ca\textsuperscript{2+}]\textsubscript{JSR} and [Ca\textsuperscript{2+}]\textsubscript{NSR}, respectively.

calmodulin kinase II (CaMKII).

The CaMKII regulation (CaMKII* in Figure 1) was implemented by adapting the mechanisms described for a canine cardiac myocyte in [10] to our model of a mouse cardiac myocyte. The CaMKII kinetics in [10] is based on the Hanson model [3]. Based on the results in [4], CaMKII should reach a certain steady-state activity level, but in the Hanson model it instead oscillates around a certain level that corresponds to the Ca\textsuperscript{2+} spiking frequency. For this reason we decided to use a more recently published CaMKII reaction scheme of Bhalla [8] that describes better the unique property of CaMKII to detect the Ca\textsuperscript{2+} spike frequency. The Bhalla model was originally designed for neurons, but its applicability for myocytes has already been documented in our previous studies [11], [12] that incorporated the enzyme part of the model.

The CaMKII regulation of ICaL, JREL and JUP is defined by the following general equation:

\[
\Delta_{\text{CaMKII}} = \Delta_{\text{MAX}} \cdot \frac{\text{CaMKII}}{\text{CaMKII} + S_{\text{half}}}
\]

where \( \Delta_{\text{MAX}} \) is the maximal CaMKII-dependent change on the target, \( \text{CaMKII} \) is the percentage of active CaMKII of the total amount of CaMKII, and \( S_{\text{half}} \) is the half-saturation coefficient.

The combined ECC and enzyme regulation model with the above mentioned features was implemented to the Matlab\textsuperscript{TM} environment of technical programming. The model is a set of deterministic differential equations, whose total number is a little over hundred. Simulation results were obtained by numerically integrating the model equations with a stiff ordinary differential equation (ODE) solver method.

4. RESULTS

The CaMKII-activity reaches a certain steady-state level that depends on the amplitude and frequency of the Ca\textsuperscript{2+} signal. In Figure 3, the steady-state activity levels are presented as function of frequency (A) and amplitude (B). In Figure 3A, the amplitude scaling factor for Ca\textsuperscript{2+} input signal is one. And in Figure 3B, the frequency of
the Ca\textsuperscript{2+} signal is constant 2 Hz. Based on these results of the dynamic range of enzyme system, we decided to use no scaling for the Ca\textsuperscript{2+} signal, i.e., the input for the enzyme reactions is the [Ca\textsuperscript{2+}] level with no multiplication. The finding that the range of enzyme activity for frequencies 1, 2 and 3 Hz was largest, when the Ca\textsuperscript{2+} signal was not scaled (data not shown), also supported this decision.

Based on the results presented in Figure 3 and extrapolation of the frequency dependence of CaMKII, the value of $S_{half}$ was set to 7.5, i.e., the maximal CaMKII-activity is expected to be about 15%.

Experiments indicate that ICaL can be increased by CaMKII regulation as much as 40% to 50% [13], [14]. Due to the limitations in the parameter adjustability of the markovian model in [7] that describes the operation of LTCC, the maximal CaMKII-dependent increase in ICaL was set to 15%. In the light of current research results, the effect of CaMKII regulation on RyR (i.e., Ca\textsuperscript{2+} release from the SR) remains controversial. While other studies have shown that CaMKII increases JREL [15], other studies give the opposite result [16]. In [10], CaMKII increases the time coefficient of inactivation, which yields a steady-state Ca\textsuperscript{2+} transient 95% greater than for control. Because CaMKII is an important mediator of the $\beta$-adrenergic signalling, whose one major function is to restore the SR Ca\textsuperscript{2+} level after a stressful change in the cell condition, we decided to choose the opposite approach. Due to the controversy, we set maximal CaMKII-dependent decrease of JREL to a moderate value of 5%.

According to experiments, CaMKII can induce a maximal increase of 70% to JUP [17]. In addition to that, the model also takes into account the disinhibitory effect of CaMKII on SERCA that is mediated by relieving the PLB inhibition. The equations describing JUP were adapted from [10] by adjusting the parameters for a mouse myocyte.

The resulting effects of rate-dependent activation of CaMKII on the three components of Ca\textsuperscript{2+} dynamics, i.e., ICaL, JREL and JUP are shown in Figure 4. As can be seen, the effects are quite moderate compared to the maximal changes that were set. This is mainly because at this phase of the model development the components, i.e., the effect of CaMKII on the three targets, had to be tested separately, which limits the use of higher pacing frequencies. The resting cardiac spike frequency of a mouse is around 5 Hz. So, some margin had to be left to the maximal effects.

5. CONCLUSIONS

We have shown that the model of ECC with the crucial corrections that we have made to Ca\textsuperscript{2+} dynamics is a suitable platform for implementation of the rate-dependent regulation. The studies on the three individual CaMKII targets have proved that the regulation mechanisms function as they should, i.e., they induce changes that are both qualitatively and quantitatively as expected.

Adjusting the parameters that define rate-dependent regulation facilitated by CaMKII is the next major task that needs to be done to verify that the model indeed is in line with current research data that is available for mouse ventricular myocytes. There is an obvious need to do experimental work as well in our research group to measure the effects of CaMKII inhibition. The model has limitations related to the markovian models that define the function of LTCC and RyR. It is likely that a more sophisticated model for LTCC is one thing that needs to be considered in the future. Experiments indicate that CaMKII induces a channel-gating mode that is characterized by frequent, long openings of the LTCCs [18]. This kind of mode change can not be implemented to the current model structure of the LTCC. A comparable solution for RyR channel-gating could be possible, but as stated above, determining the effect of CaMKII regulation on RyR requires more experimental research.

The added complexity of the model will obviously decrease computational efficiency. This raises the question, what can be done to keep the model complexity at a level that still allows it to be run on a desktop computer
with reasonable computation times. Already with the current model implementation this is a hot topic. In the near future, we have to find some solutions to these challenges. At least two approaches are obvious. First is the use of more efficient computational tools for either solving the differential equations [19] or reducing the differential equation system [20]. The other way would be to reduce the intrinsic complexity of the system by using the kind of simplifications that have been done, e.g., for the markovian models to derive a computationally efficient implementation of CICR model in [21].

6. REFERENCES
ABSTRACT

We use topographic visualisation to investigate microarray gene expressions for breast cancer prognosis by performing unsupervised, model-free data mining. For different subsets of gene expressions we find similar visualisation maps, indicating it is the relative distributions across many genes which is more significant for breast cancer prognosis than the expression level of a specific subset of genes.

1. INTRODUCTION

Breast cancer death occurs principally from the primary tumour metastases at distant sites. The clinical heterogeneity of breast cancer and varying and patient-specific response to treatment points to a genetic complexity and a systems-level issue of the interaction between genetic profiles and environmental factors. From a systems biology perspective, it is therefore clear why attempts to extract low dimensional prognostic markers for breast cancer – either gene predictors or traditional biomarkers are limited. Many women receive unnecessary cytotoxic chemotherapy unnecessarily due to the weak performance of current prognostic techniques. In this paper we examine a microarray breast cancer dataset using techniques from unsupervised topographic visualisation with the aim of investigating whether specific gene subsets can be used as prognostic indicators.

1.1. Data

We used a subset of the data used in the well-known study of van’t Veer et.al.[1] in which we focus on 78 sporadic lymph-node negative patients. Of these 78 patients, 34 developed distant metastases within 5 years and 44 remained free of cancer in that period. These are regarded as poor and good prognosis groups respectively. The interest is whether the information contained in a gene expression profiling could provide a good prognosis of clinical outcome into poor and good prognosis groups. In the van’t Veer study, from an initial set of approximately 25000 human genes synthesised by inkjet microarray technology, about 5000 were found to be significantly expressed. These were further reduced to 70 genes by a process to select genes to maximise classification. The eventual classifier correctly predicted 83% of the patients into the prognosis groups with 5 poor prognosis and 8 good prognosis patients classified into the opposite categories. Hence it was concluded that these 70 genes provided good predictors of breast cancer outcome. This of course is not a genuine out-of-sample measure of classification accuracy and so can be regarded as an upper bound to performance.

In a separate recurrence study [2] to predict nodal metastasis states and recurrence, Huang et.al. came to a similar conclusion, that DNA microarray data provides prognostic information for breast cancers, though they emphasised the complexity and interrelated nature of the various biological processes likely underpinning breast cancer recurrence.

The use of gene subsets as prognostic factors remains controversial. In recent work [3], it was shown that the van’t Veer 70 gene set was no more effective at prognosis that the Nottingham Prognostic Indicator (NPI) or a suitably trained artificial neural network using traditional non-genomic biomarkers. This is not surprising from a systems biology perspective, where we would regard cancer as the result of complex interactions between genetic, biological and environmental influences.

The Milan group [4] has re-analysed the van’t Veer data and produced a larger related list of 641 genes as a starting basis for data mining. For the remainder of this paper we start with the Milan Group’s list of genes for these 78 patients.

1.2. Visualisation

Our aim is not to build supervised classification models, but to introspectively data mine the van’t Veer microarray information in an unsupervised manner, to examine structure and relationships in the high dimensional data.

Algorithms for reduced dimensionality representation have previously been used to visualise microarray data beyond the standard and limiting dendrogram. For instance, the Self Organising Map (SOM) has been used to investigate yeast [5] and human cancers [6] and [7], the latter
in combination with the $k$-means algorithm. Analogously, Principal Component Analysis (PCA) has been used to investigate yeast [8] and to identify tissue-specific expression of human genes [9]. However, both SOM and PCA have significant drawbacks. PCA is a variance-preserving linear projection, and this limitation does not necessarily lead to an optimal topographic representation [10]. On the other hand, SOM lacks a sound theoretical underpinning (for example, there is no cost function to optimise, and training parameters must be chosen arbitrarily).

A software package, called MILVA (Microarray Latent Visualization and Analysis [11]), has been developed for the purpose of low dimensional topographic data visualisation. The aim of MILVA is to overcome some of the limitations of less flexible clustering methods by involving the direct interaction of the biologist in detecting patterns across co-regulated genes. One of the tools inside MILVA is the NeuroScale model, which we use here.

1.3. Neuroscale

In a Neuroscale [12] topographic map the distribution and relative positions of the points in the data space are determined to reflect the relative dissimilarity between data measurements (gene expression values) in the high-dimensional space, and hence generalises the established Sammon map. $N$ measurement vectors $x_i$ in $\mathbb{R}^p$ are transformed using a Radial Basis Function (RBF) network to a corresponding set of feature (visualisation) vectors $y_i$ in $\mathbb{R}^q$. Generally, $q < p$ as dimension reduction is desired, and typically $q = 2$ for visualisation. The quality of the projection is measured by the Sammon stress metric (n.b. we are using a reduced form here, neglecting a denominator often employed):

$$ E = \sum_{i=1}^{N} \sum_{j=1}^{N} (d_{ij}^* - d_{ij})^2, $$

where $d_{ij}^* = \|y_i - y_j\|$ and $d_{ij} = \|x_i - x_j\|$ represent the inter-point distances in projection space and data space respectively.

This approach has been extended to allow the incorporation of dynamics as in temporally ordered microarray experiments [13], and to incorporate quantitative uncertainty in data [14]. However for this paper we use the vanilla NeuroScale model.

We now use the NeuroScale model to visualise and interrogate our data set, considered as an array of $78 \times 641$ patterns. Note, we can treat this either as 78 patterns of dimension 641 and hence visualise patients or treat it as 641 patterns of dimension 78, allowing us to visualise genes. We do both here.

2. RESULTS

We visualise the data using the van’t Veer list of 70 genes, our list of 70 genes and 176 genes obtained by simple selection using MILVA and small genes selections.

2.1. van’t Veer data

Figure 1 shows the NeuroScale plot of the van’t Veer gene choice. Note that the poor and good prognosis groups are visually separated with the exception of a small number of patients. By imagining a simple classifier separating line in this figure 4 of the good prognosis and 12 of the poor prognosis patients would be misclassified. This supports the work done in [1]. It is also clear that the misclassifications are well dispersed (ie are not ‘borderline’ cases).

![Image 1](image1.png)

Figure 1. NeuroScale map of the van’t Veer gene set. Note the broad separation between poor (grey diamonds) and good (black circles) prognosis groups.

2.2. Alternative gene set.

We used the MILVA tool to select 70, and then 176 genes for comparative visualisation purposes. We constructed a ranked list of genes from each prognosis group with the smallest variance across patients up to a threshold variance. We then selected those genes in one group which were not in the list for the other group. (ie we selected the top 35 or 88 genes from the weak prognosis group which were not in the good prognosis group, added to the 35 or 88 from the other group with smallest variance). This was not a process selected for any medical realism — merely to demonstrate the collective patterns across gene space. However, our motivation was that genes with small variance across patients have consistent response across patients. So these genes would have less noise distorting the underlying biology. To emphasise though, they are not chosen to explicitly maximise discrimination.

Figure 2 shows the NeuroScale visualisation of the 70 gene set and 3 shows the corresponding view using the 176 gene set.

Although we have symbol coded the different prognosis groups (black circles and grey diamonds), this is merely for visualisation purposes, and no class information was used in the NeuroScale Algorithm. These displays can be considered to be unsupervised. However it is clear that a reasonable separation exists between prognosis groups. In the 176 case, a simple linear classifier
would misclassify 11 good prognosis patients and 1 or 2 poor prognosis patients. This is the more appropriate sensitivity-specificity since we would wish to classify more of the poor prognosis patients correctly to identify appropriate care regimes. However, to emphasise, this is not a classification model – it is a data visualisation and investigation model.

Hence, in terms of visualisation, the van’t Veer set and this set give comparable visualisations in terms of distributions of patients. However, note that if we compare the two 70-gene lists, there are only 5 genes which are in common. i.e. we are producing a similar visualisation map with an almost complementary gene list. This implies that there is nothing unique in the van’t Veer 70 gene list for prognostic specialisation through visualisation. It is possible that the visualisation STRESS cost function might have missed something important from a classification perspective, but this should then present strong differences in the visualisations since differences in classification ability would reflect a strong dissimilarity between gene expression profiles. We interpret this result of non- uniqueness in terms of the clinical heterogeneity of gene–biology–environment interactions. We therefore regard this lack of uniqueness as implying that it is the pattern of activity across the broad gene set which reflects patient group specialisation. The gene-gene similarities indicate a distribution of gene response patterns and it is this distribution that is important, rather than the expression of a specialist unique gene subset. Different genes may be involved in different circumstances to differing degrees. We are going to be interested in that pattern of activity in the future.

We can see this in more detail if we now perform a gene-specific NeuroScale visualisation rather than a patient-specific map. Figures 4 and 5 show the projections of the poor and good prognosis groups respectively. Now, each
point in the map represents a gene. We have numbered the genes in the maps for convenience. What we observe is that the gene response patterns have a spiculated distribution: the local density of gene similarity across the patients has a distinctive structure – it is not random! The distance between genes in this map reflects how dissimilar the response of the genes are across patients. The 70 gene maps are quite similar to these 176 gene maps.

3. CONCLUSION

We have applied a topographic visualisation approach to the analysis and discussion of high dimensional DNA microarray data in the context of breast cancer prognosis. From a systems biology perspective, we infer that it is not the expression of specific genes that is relevant, but the relative distribution patterns of a large range of genes that is important. This distribution is likely to be influenced by environmental factors (ie. patient-specific factors beyond genomics) and so gene expression profiles alone are not likely to be dominant as prognostic indicators.

4. ACKNOWLEDGEMENTS

We thank the Milan group, and especially Nicola Lama and Elia Biganzoli for valuable discussion and interaction, and for supplying us with details on their analysis of the van’t Veer data. We also thank Davide d’Alimonte for developing the MILVA tool for investigating microarray data. This work was partially supported by the BBSRC contract 92/EGM17737 and the EU BIOPATTERN Network of Excellence, under contract 508803.

5. REFERENCES


ABSTRACT

We have previously shown how data from high dimensional and time-course gene expression microarray slides can be usefully explored using data visualisation techniques based on topographic projections, and in particular a method known as NeuroScale. However, the method does not scale well with numbers of data points. In this paper we introduce a hierarchical NeuroScale architecture which does not assume any prior model on the data, and which circumvents this computational complexity issue. An automatic partitioning is produced in which each partition is a latent NeuroScale map. These separate latent maps are connected by a linear transformation which allows a single visualisation map to be produced. The application of this method to gene expression data is used to show the effectiveness of the algorithm.

1. INTRODUCTION

A major emerging aspect of computational systems biology is the use of automated software tools for prospective data mining. The difficulty though, stems from the very high dimensional nature of data measurements, coupled with the nonlinearity, lack of strong prior models, nonstationary nature of dynamics in biology, and the often very noisy nature of the obtained data.

In a recent review on “Statistical Challenges in Functional Genomics”, Sebastiani et al. [1] commented, “The newly born functional genomic community is in great need of tools for data analysis and visual display of the results”. Such tools in the clinical/biomedical domains have tended to rely upon clustering and the dendrogram, Khan et. al. [2], or projective methods such as Principal Component Analysis, Raychaudhuri [3], and Sammon map, Apostol and Szpankowski [4], and Independent Component Analysis, Draghici [5]. Other recent methods for data mining and knowledge discovery in databases have developed using minimal spanning trees for complex data visualisation, Laskaris [6], capable of patient-specific and test-specific analysis.

Tools for data exploration, especially through low dimensional data visualisation are especially important for users such as research scientists or clinicians who are not specialists in data modelling. Visualisation is an effective way for domain experts to detect trends, structures, clusters, outliers, and other important data characteristics. In addition, it can be used to guide the data analysis process by giving feedback on the results of models.

We have recently emphasised the topographic requirement of visualising high dimensional data, and in particular developed the NeuroScale [7] approach to data visualisation for gene microarrays [8, 9, 10].

However, because NeuroScale employs a similarity measure between any two data elements to motivate the topographical measure, the training will be very expensive when a large data set is given since the number of the distances scales as $N(N-1)/2$. In addition, due to the complexity of the data structure, sometime one global latent map cannot give a very good latent model for all the data.

In this paper we propose an algorithm which circumvents this difficulty by employing a hierarchical structure. First, the data set can be partitioned into several subsets. For each subset, the NeuroScale algorithm is applied and a latent map is established. By including all the cluster centres in each subset, the relative positions of the subset in global latent space can be determined and a linear transformation between any two subsets may be obtained. By utilising these transformations a single global latent map can be established by transforming between various local maps. The proposed hierarchical method can both reduce the training time and reveal some hidden structures where the classical NeuroScale algorithm has difficulty.

2. HIERARCHICAL NEUROSCALE

2.1. The principle of NeuroScale

Assume $x_i = (x_{i1}, \ldots, x_{id}) \in \mathbb{R}^d, i = 1, 2, \ldots, N$ are in high dimensional space $d$. If we want to look at the interrelationships in the data, we can project each data from this space into a new but low dimensional space as a new point $y_i = (y_{i1}, \ldots, y_{ip}) \in \mathbb{R}^p$ (Normally $p$ equals two or three dimensions) with $d > p$ by using the mapping

$$Y = JW$$
where $Y$ is a matrix of a row latent vectors $y_i$, $W$ is the weight matrix and $J$ is the matrix defined as below

$$J_{ij} = \phi_j(x_i)$$

where $\phi_i, i = 1, \ldots, M$ are the basis functions.

The NeuroScale approach determines the above basis functions as well as the weight vector $W$ so as to minimise the following nonlinear STRESS criterion

$$E = \sum_{i=1}^{N} \sum_{j=1}^{N} (d_{ij}^* - d_{ij})^2$$

with $d_{ij}^* = \|x_i - x_j\|$, $d_{ij} = \|y_i - y_j\|$. The inter-point distances in the latent visualisation space will then match as closely as possible the inter-point similarity distances in the original high dimensional data space, and so structure in the original space is preserved in the visualisation space. This is not the case for projective techniques such as PCA or dendrograms.

### 2.2. Hierarchical NeuroScale

NeuroScale can use any kind of semiparametric nonlinear mapping function to establish a projection. However, normally a Radial Basis Function network is used. One attraction of the class of RBF networks is that there is a two-stage training procedure which is considerably faster than the methods used to train multi-layer perceptrons. In the first stage, the basis function parameters are set so that they model the unconditional model density. In the second stage, the neural network weights are determined. The speed of the training RBF networks makes them attractive for use in the NeuroScale.

In the classical NeuroScale model realised using RBF networks, if $K$ basis functions are to be used, a K-mean clustering method is applied to seek $K$ clusters and position each basis function on one cluster centre. Extending this idea further, we split the data set into a certain number of subsets by using K-means. For each subset we establish a separate NeuroScale latent map.

The K-mean method is applied to partition the dataset by minimising:

$$E = \sum_{j=1}^{K} \sum_{n \in S_j} \|x_n - \mu_j\|^2$$

where $K$ is the number of clusters, $\mu_j$ is the centre of the $j$th cluster which is given by

$$\mu_j = \frac{1}{N_j} \sum_{k \in S_j} x_k$$

with $N_j$ the number of the data points in the cluster $j$.

The mapping error is modified so that the cluster centre is given the dominant significance by weighting the cluster centre with $\delta_j > 1$ and in the extreme situation, we can set $\delta_j = N_j$. For example, when training cluster $S_j$, the criterion may be

$$E = \sum_{n=1}^{N} \sum_{l>n} (d_{nl}^* - d_{nl})^2 + \sum_{n=1}^{N} \delta_n (d_{nl}^* - \bar{d}_{nl})^2$$

where $d_{nl}^* = \|x_n - \mu_l\|$, $d_{nl} = \|y_n - y_{\mu_l}\|$, $\bar{d}_{nl}^* = \|\mu_n - \mu_l\|$, $\bar{d}_{nl} = \|y_{\mu_n} - y_{\mu_l}\|$, $y_{\mu_l}$ is the projection of $\mu_l$ in the low dimensional latent space.

#### 2.2.1. Sub-clusters relocation and the single map establishment

The latent map obtained by visualising each subset separately will not allow us to see the overall positions of the whole data. To do this, we need to combine all the sub-maps into a single global map.

To do so, we can linearly transform all the sub-cluster latent maps to new positions relatively related to their high-dimensional positions. To fulfil the task, only rotations, translations, and reflections are allowed to be used to keep the relative data position in each latent map unchanged. To get this linear transform, the cluster centre positions in each of the latent spaces are used. For the $K$ cluster centres $\mu_{i,k}, i = 1, 2, \ldots, K$ in the original data space, there will be $K$ cluster centres in each of the latent spaces. We assume these centres are $\nu_{jk}, k = 1, 2, \ldots, K$ for the subset $S_j$. Then for any two different subsets $S_i, S_j$, normally we will have $\nu_{jk} \neq \nu_{jk}, k = 1, 2, \ldots, K$. Assume every centre is mapped perfectly in the latent space, that is they all keep their relative distances exactly as they were in the original data space. Then by using a series of translations, rotations, and reflections, they can be matched exactly in the latent space. Obviously, this assumption is not precisely true. Consider the mapping error. First, we select a subset cluster which minimises the following criterion:

$$J_i = \frac{1}{K} \sum_{p=1}^{K} \sum_{q>p} (d_{pq}^* - d_{\nu(p,q)})^2$$

for all $i \in \{1, 2, \ldots, K\}$. Without loss of generality, we assume this subset is $i = 1$. Then the latent map of the subset 1 is used as the anchor map. When matching subset $j, j \neq 1$, first we translate both $\nu_{1j}$ and $\nu_{jj}$ to the origin of the latent space by using:

$$\nu_{1k} = \nu_{1k} - \nu_{1j}$$

$$\nu_{jk} = \nu_{jk} - \nu_{jj}$$

for all $k = 1, 2, \ldots, K$. Secondly, we seek a linear transform $A$ which minimises the following criterion:

$$J_2 = \frac{1}{K} \sum_{k=1}^{K} (A\nu_{jk} - \nu_{1k})^2$$

Where $A$ is only allowed to be a rotation

$$A = \left( \begin{array}{cc} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{array} \right)$$
and a reflection plus a rotation.

\[ A \equiv \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \]  

(9)

By replacing these two transforms into (7) respectively, we can get

\[ \tan \theta = \frac{\sum_{k=1}^{K} (\hat{v}_{1k1} \hat{v}_{jk2} - \hat{v}_{1k2} \hat{v}_{jk1})}{\sum_{k=1}^{K} (\hat{v}_{1k1} \hat{v}_{jk1} + \hat{v}_{1k2} \hat{v}_{jk2})} \]  

(10)

\[ \tan \theta = \frac{\sum_{k=1}^{K} (\hat{v}_{1k1} \hat{v}_{jk2} + \hat{v}_{1k2} \hat{v}_{jk1})}{\sum_{k=1}^{K} (\hat{v}_{1k1} \hat{v}_{jk1} - \hat{v}_{1k2} \hat{v}_{jk2})} \]  

(11)

where \( \hat{v}_{ij} = (\hat{v}_{ij1}, \hat{v}_{ij2}) \). Then for each equation, two different \( \theta \) values can be obtained as

\[ \theta_1 = \arctan \left( \frac{\sum_{k=1}^{K} (\hat{v}_{1k1} \hat{v}_{jk2} - \hat{v}_{1k2} \hat{v}_{jk1})}{\sum_{k=1}^{K} (\hat{v}_{1k1} \hat{v}_{jk1} + \hat{v}_{1k2} \hat{v}_{jk2})} \right) \]  

(12)

\[ \theta_2 = \pi + \arctan \left( \frac{\sum_{k=1}^{K} (\hat{v}_{1k1} \hat{v}_{jk2} - \hat{v}_{1k2} \hat{v}_{jk1})}{\sum_{k=1}^{K} (\hat{v}_{1k1} \hat{v}_{jk1} + \hat{v}_{1k2} \hat{v}_{jk2})} \right) \]  

(13)

\[ \theta_3 = \arctan \left( \frac{\sum_{k=1}^{K} (\hat{v}_{1k1} \hat{v}_{jk2} + \hat{v}_{1k2} \hat{v}_{jk1})}{\sum_{k=1}^{K} (\hat{v}_{1k1} \hat{v}_{jk1} - \hat{v}_{1k2} \hat{v}_{jk2})} \right) \]  

(14)

\[ \theta_4 = \pi + \arctan \left( \frac{\sum_{k=1}^{K} (\hat{v}_{1k1} \hat{v}_{jk2} + \hat{v}_{1k2} \hat{v}_{jk1})}{\sum_{k=1}^{K} (\hat{v}_{1k1} \hat{v}_{jk1} - \hat{v}_{1k2} \hat{v}_{jk2})} \right) \]  

(15)

We choose the one which minimises \( J_2 \). All the data \( x_i \) in the subset \( j \) are then transformed by

\[ \hat{y}_i = A(y_i - \nu_{ij}) + \nu_{ij} \]  

(16)

3. USING HIERARCHICAL NEUROSATELE FOR VISUALISE GENE EXPRESSION DATA

*S. coelicolor* is a complex mycelial Gram positive bacterium which undergoes developmental changes leading ultimately to sporulation and production of antibiotics and other secondary metabolites. The 7825 predicted genes in the linear *S. coelicolor* chromosome include more than 20 gene clusters coding for known or predicted secondary metabolites. The genome also contains an unusually large proportion of regulatory genes [8].

The *S. coelicolor* dataset was produced and provided by Surrey University 1 and consists of expression data from samples of surface-grown cultures taken at 16, 18, 20, 21, 22, 23, 24, 25, 39 and 67 hours after the inoculation of the growth medium. Two independent sets of cultures were precessed and each Cy3-labelled cDNA sample was hybridised against Cy5-labelled DNA (gDNA) as the common reference. The signal in the gDNA channel follows a different distribution from the cDNA channel since the number of copies of the gene in the genome is fixed while the number of RNA copies can vary widely. There are many probes yielding a significant gDNA signal and vary little signal in the cDNA channel, resulting in a long (left hand) tail in the distribution of ratios.

The preprocessing of the microarray data consisted of: correcting the data for spatial effects; taking the log-ratio of the signal and the reference measurements; applying the quantile method for cross-condition normalisation; applying a variance filtering and a low-value filtering to remove genes that are not significantly expresses and finally normalising each pattern of gene expression by subtracting the values at the first time step.

The analysis in this paper is only based on those genes having a significant level of expression, and here 3061 genes are used [8].

The visualisation results using the proposed hierarchical NeuroScale algorithm are shown in Figure 1, where the data set is partitioned into ten subsets and each one is visualised separately and the results shown in the lower row figures. The upper figure is the combination result. From this single figure the relative position of all the subsets can be seen. Figure 2 [8] is a visualisation result obtained by training the whole data set and shaded to match the same partitions as in Figure 1. Obviously the match is good, by comparing Figure 1 and 2. From this example we can also see from Figure 1, and 2 that there are some clusters overlapping. But the hierarchical NeuroScale algorithm has separated them very well by mapping them into different sub-maps. Figure 3 shows the time series display of a special cluster.

4. CONCLUSION

This paper has introduced a new method for hierarchical data visualisation of high dimensional data. We have demonstrated its implementation on time-course gene expression microarray profiles of *S. coelicolor*, where we visualised 3061, ten-dimensional patterns. The algorithm named Hierarchical NeuroScale proposed in this paper has several advantages over the classical NeuroScale architecture. In turn, NeuroScale has several advantages over other ‘classical’ visualisation methods [7]. By clustering the whole dataset into a number of subclusters and visualising each cluster in a linked set, the requirement for computer memory can be reduced as well as the computing time. By combining all the latent maps together, the relative positions of data can be maintained. In addition, by using the k-means clustering method, some hidden structures can be found automatically. The experimental results of gene expression data used in this paper confirm the above advantages. From these results we can see the method has the capability to group the genes with similar profiles together. In addition, co-expressed genes will be in the same child map. This hierarchical NeuroScale algorithm can be used to visualise gene expression data with more than 10 dimensions. In other studies we are unable to report here, we have used this new algorithm to visualise many patterns of several hundred dimensions.

---

1http://www.surrey.ac.uk/SBMS/Fgenomics/
Figure 1. The Bottom-up Hierarchical NeuroScale visualisation result. Upper: the integrated visualisation result for all the data. This map is obtained by combining the 'child' maps shown below the integrated map. Lower: the subset visualisation results. The training used all the data points in each of the subsets. The highlighted points in the upper map and the lower map relate to the same points. From these points we can see that three clusters are overlapped in the integrated map.

5. ACKNOWLEDGEMENTS

We are indebted to Colin Smith and his team at Surrey University, UK for providing the data.

6. REFERENCES


SIMULATION STUDY OF DETERMINISTIC DIFFERENTIAL EQUATION MODEL FOR PROTEIN KINASE C SIGNALING: SENSITIVITY OF STIMULI, PARAMETER VALUES, AND INITIAL CONCENTRATIONS

Tiina Manninen\textsuperscript{1,2}, Antti Saarinen\textsuperscript{1}, Keijo Ruohonen\textsuperscript{2}, and Marja-Leena Linne\textsuperscript{1}

\textsuperscript{1}Institute of Signal Processing, Tampere University of Technology, 
\textsuperscript{2}Institute of Mathematics, Tampere University of Technology, 
P.O. Box 553, FI-33101 Tampere, Finland 
tiina.manninen@tut.fi, antti.saarinen@cs.tut.fi, keijo.ruohonen@tut.fi, marja-leena.linne@tut.fi

ABSTRACT
Due to the complexity of biophysical and biochemical processes, as well as the increasing availability of experimental data, considerable attention has been given to the mathematical modeling and the simulation of nerve cell functions. In this work, we study the sensitivity of different types of stimuli, parameter values, and initial concentrations using a differential equation model of an intracellular protein kinase C signal transduction pathway. To better understand complex neuronal behaviors, it is important to model and simulate intracellular events, including the signal transduction pathways.

1. INTRODUCTION
Recent studies provide evidence that regulation of intracellular phenomena is important for transitions between different physiological and behavioral states of an organism (see e.g. \cite{1}). Several components of the cell membrane, such as ion channels and receptors, are regulated intracellularly, for example by protein kinases. One commonly known kinase is the protein kinase C (PKC). It belongs to a family of enzymes that are known to exist in all cell types and participate in complex signal transduction pathways \cite{2}. In neurons, the ion channel - PKC association has been shown to underly excitability changes \cite{1}. Furthermore, the activation of PKC by intracellular calcium (Ca\textsuperscript{2+}), induced by N-methyl-D-aspartate (NMDA) receptors, may be important in the generation of long-term potentiation (LTP) \cite{3}. Overall, PKC is suggested to contribute to several types of neuronal plasticity, including LTP \cite{4}, and thereby to participate in forming a biochemical basis for short-term memory \cite{5}. On the other hand, abnormal concentrations of PKC may play a role in neurogenesis, as well as in degeneration of neurons.

There are only a few reaction kinetic models developed for neuronal PKC signaling. One model is presented for the hippocampal neuron \cite{5} and another one for the cerebellar Purkinje cell \cite{6}. In order to develop a similar PKC model for the cultured cerebellar granule neuron, we here perform a sensitivity analysis for one of the previously presented models \cite{5} and analyze its suitability for our studies. This work is a summary of larger report presented in \cite{7}.

2. METHODS

2.1. PKC signal transduction pathway
The model describing the PKC pathway is obtained from the Database of Quantitative Cellular Signaling (DOQCS) \cite{5}, \cite{8}. The database contains model structures (interactions), rate constants, and initial concentrations including annotations and information on data sources.

PKC isoymes can be categorized into three subclasses, conventional, novel, and atypical PKC. The PKC model used in our study (Fig. 1) represents the conventional PKC which means that it is a weighted average of $\alpha$, $\beta$, and $\gamma$ isoforms \cite{2}. Conventional PKC is regulated by the following second messengers: Ca\textsuperscript{2+}, diacylglycerol (DAG), and phosphatidylserine (PS) or arachidonic acid (AA) (INPUTS in Fig. 1) \cite{2}. The PKC model consists of ten reversible reactions ($R_j$, $j = 1,...,10$) and 15 different interacting chemical species of which 11 are model variables. Chemical species with asterisks (*) in Figure 1 denote computational intermediates which enable the study of the concentrations before they are summed to active PKC (PKCa; OUTPUT in Fig. 1).

![Figure 1. Graphical design model of the PKC pathway, which was originally published in Bioinformatics \cite{9}.](image-url)
Table 1. Reversible reactions, reaction rates, and rate constants for PKC pathway model [7, 8].

<table>
<thead>
<tr>
<th>Reaction</th>
<th>Reaction rate</th>
<th>$k_1$</th>
<th>$k_{-1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1 PKCi $\xrightarrow{k_{11}^{R1}}$ PKCbasal*</td>
<td>$v_{R1} = k_{11}^{R1}[PKCi] - k_{11}^{R1}[PKCbasal^*]$</td>
<td>1/2</td>
<td>50/2</td>
</tr>
<tr>
<td>R2 PKCi + AA $\xrightarrow{k_{12}^{R2}}$ AAPKC*</td>
<td>$v_{R2} = k_{12}^{R2}[PKCi][AA] - k_{12}^{R2}[AAPKC^*]$</td>
<td>120 $\frac{1}{5\pi}$</td>
<td>0.1 $\frac{1}{2}$</td>
</tr>
<tr>
<td>R3 CaPKC $\xrightarrow{k_{13}^{R3}}$ CaPKCmemb*</td>
<td>$v_{R3} = k_{13}^{R3}[CaPKC] - k_{13}^{R3}[CaPKCmemb^*]$</td>
<td>1.2705 $\frac{1}{5\pi}$</td>
<td>3.5026 $\frac{1}{2}$</td>
</tr>
<tr>
<td>R4 CaPKC + AA $\xrightarrow{k_{14}^{R4}}$ AACaPKC*</td>
<td>$v_{R4} = k_{14}^{R4}[CaPKC][AA] - k_{14}^{R4}[AACaPKC^*]$</td>
<td>1200 $\frac{1}{5\pi}$</td>
<td>0.1 $\frac{1}{2}$</td>
</tr>
<tr>
<td>R5 DAGCaPKC $\xrightarrow{k_{15}^{R5}}$ DAGCaPKCmemb*</td>
<td>$v_{R5} = k_{15}^{R5}[DAGCaPKC] - k_{15}^{R5}[DAGCaPKCmemb^*]$</td>
<td>1 $\frac{1}{2}$</td>
<td>0.1 $\frac{1}{2}$</td>
</tr>
<tr>
<td>R6 AADAGPKC $\xrightarrow{k_{16}^{R6}}$ AADAGPKC*</td>
<td>$v_{R6} = k_{16}^{R6}[AADAGPKC] - k_{16}^{R6}[AADAGPKC^*]$</td>
<td>$2 \frac{1}{2}$</td>
<td>0.2 $\frac{1}{2}$</td>
</tr>
<tr>
<td>R7 PKCi + Ca$^{2+}$ $\xrightarrow{k_{17}^{R7}}$ CaPKC</td>
<td>$v_{R7} = k_{17}^{R7}[PKCi][Ca^{2+}] - k_{17}^{R7}[CaPKC]$</td>
<td>6 $\cdot 10^7$ $\frac{1}{5\pi}$</td>
<td>0.5 $\frac{1}{2}$</td>
</tr>
<tr>
<td>R8 CaPKC + DAG $\xrightarrow{k_{18}^{R8}}$ DAGCaPKC</td>
<td>$v_{R8} = k_{18}^{R8}[CaPKC][DAG] - k_{18}^{R8}[DAGCaPKC]$</td>
<td>7999.8 $\frac{1}{5\pi}$</td>
<td>8.6348 $\frac{1}{2}$</td>
</tr>
<tr>
<td>R9 PKCi + DAG $\xrightarrow{k_{19}^{R9}}$ DAGPKC</td>
<td>$v_{R9} = k_{19}^{R9}[PKCi][DAG] - k_{19}^{R9}[DAGPKC]$</td>
<td>600 $\frac{1}{5\pi}$</td>
<td>0.1 $\frac{1}{2}$</td>
</tr>
<tr>
<td>R10 DAGPKC + AA $\xrightarrow{k_{20}^{R10}}$ AADAGPKC</td>
<td>$v_{R10} = k_{20}^{R10}[DAGPKC][AA] - k_{20}^{R10}[AADAGPKC]$</td>
<td>$1.8 \cdot 10^4$ $\frac{1}{5\pi}$</td>
<td>2 $\frac{1}{2}$</td>
</tr>
</tbody>
</table>

2.2. Mathematical formulation

Each reversible reaction shown in Figure 1 can be given using the law of mass action. In the reaction

$$A + B \xrightleftharpoons{v}{k_1}{k_{-1}} C,$$

(1)

A is the reacting species, B is the second messenger, and C is the product species. $k_1$ is the forward and $k_{-1}$ the backward reaction constant. The reaction presented in (1) can be described as a differential equation of the form

$$v = \frac{d[C]}{dt} = -\frac{d[A]}{dt} = k_1[A][B] - k_{-1}[C],$$

(2)

where $v$ is the reaction rate for the reversible reaction in (1), and [A], [B], and [C] are the concentrations of A, B, and C, respectively. In this formulation, the interactions are solely specified by the rate constants, the concentrations of the second messengers, and the initial values of the variables. In the case of a system of differential equations, the equation for every variable $[X_i]$ can be written in the form

$$\frac{d[X_i]}{dt} = \sum_j s_{ij}v_j,$$

(3)

where $s_{ij}$ is the stoichiometric coefficient and $v_j$ is the reaction rate for the reversible reaction $j$. The algebraic equation summing the effect of computational intermediates to the concentration of PKCa is of the form

$$[PKCa] = [PKCbasal^*] + [AAPKC^*] + [CaPKCmemb^*] + [AACaPKC^*] + [DAGPKCmemb^*] + [AADAGPKC^*].$$

Table 2. Reaction rates and initial concentrations for each variable of PKC pathway model [7, 8].

<table>
<thead>
<tr>
<th>Reaction for each variable</th>
<th>Initial concentration</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d[PKCa]/dt$ = $-v_{R1} - v_{R2} - v_{R3} - v_{R4} - v_{R5}$</td>
<td>$10^{-6}$ M</td>
</tr>
<tr>
<td>$d[CaPKC]/dt$ = $v_{R7} - v_{R8} - v_{R9} - v_{R10}$</td>
<td>$3.7208 \cdot 10^{-23}$ M</td>
</tr>
<tr>
<td>$d[DAGPKC]/dt$ = $v_{R9} - v_{R10} - v_{R11}$</td>
<td>$1.161 \cdot 10^{-22}$ M</td>
</tr>
<tr>
<td>$d[CaPKCmemb^*]/dt$ = $v_{R8} - v_{R9}$</td>
<td>$8.4632 \cdot 10^{-29}$ M</td>
</tr>
<tr>
<td>$d[AADAGPKC^*]/dt$ = $v_{R10} - v_{R11}$</td>
<td>$2.5188 \cdot 10^{-25}$ M</td>
</tr>
<tr>
<td>$d[PKCbasal^*]/dt$ = $v_{R1}$</td>
<td>$2 \cdot 10^{-8}$ M</td>
</tr>
<tr>
<td>$d[AAPKC^*]/dt$ = $v_{R2}$</td>
<td>$1.8133 \cdot 10^{-23}$ M</td>
</tr>
<tr>
<td>$d[CaPKCmemb^*]/dt$ = $v_{R3}$</td>
<td>$1.3896 \cdot 10^{-23}$ M</td>
</tr>
<tr>
<td>$d[DAGPKC^*]/dt$ = $v_{R4}$</td>
<td>$1.75 \cdot 10^{-22}$ M</td>
</tr>
<tr>
<td>$d[DAGPKCmemb^*]/dt$ = $v_{R5}$</td>
<td>$9.4352 \cdot 10^{-27}$ M</td>
</tr>
<tr>
<td>$d[AADAGPKC^*]/dt$ = $v_{R6}$</td>
<td>$4.9137 \cdot 10^{-24}$ M</td>
</tr>
</tbody>
</table>

The complete system of Figure 1 comprises ten reversible reactions in Table 1 and the algebraic equation (4). According to (2) and (3), the reversible reactions of the PKC pathway are described by the differential equations in Tables 1 and 2. The reversible reactions in Table 1 are the basis for implementing the PKC pathway into simulation tools (see also [9]). In some tools, each reaction can be implemented separately meaning that forward and backward reactions are separated. The reaction rates in Table 1 can also be used in implementation. Table 2 contains the differential equations for 11 variables. The equations for every variable in Table 2 are the basis for computing the time-series behavior of the PKC pathway. These can also be used in implementation.
2.3. Model implementation

GENESIS neuronal simulation tool [10] can be extended by Kinetikit graphical user interface (GUI) [11, 10] to simulate the activities of large networks of interacting species. In Kinetikit, a particular script language or GUI can be used to implement networks. A survey was made to compare various freely available simulation tools [9]. GENESIS/Kinetikit simulation tool was chosen for this study since Kinetikit provides a convenient way of introducing external perturbations for networks. Furthermore, there is a need for integrating different kinds of models (time and space aspects) at different levels of detail to study the cellular functions and GENESIS/Kinetikit is one of the few available tools.

In this study, versions 2.2 and 2.2.1 of GENESIS simulation tool and versions 9 and 10 of Kinetikit are used. GENESIS/Kinetikit simulation tool runs on UNIX and Linux platforms. The PKC pathway model is implemented using Kinetikit GUI. The implementation of models involves the following: (1) identifying the chemical species of the PKC pathway (see Section 2.1, Fig. 1); (2) defining the interactions between the chemical species (see Section 2.1, Fig. 1); (3) setting up the reactions using basic chemical reaction kinetics (see Section 2.2, Table 1) or differential equations (see Section 2.2, Tables 1 and 2); (4) setting the initial concentrations of the chemical species (see Table 2); and (5) setting up the stimuli (INPUTS in Fig. 1) using GUI or loading in a file containing the stimulus protocol. The initial values of the variables can be given as concentrations or as numbers of chemical species. Computations in Kinetikit are done using numbers of species.

3. SIMULATION RESULTS

The behavior of the PKC pathway is simulated using different kinds of numerically generated stimuli for Ca$^{2+}$, AA, and DAG, such as linear and step functions as well as rectangular and sine waves. The maximum free intracellular [Ca$^{2+}$], excluding the Ca$^{2+}$ stores, is 25 μM [12], but it may vary from the resting concentration of 100 nM to 10 μM [13]. For AA and DAG, we are limited to only a few experimental findings [5, 6], in which 10 μM or 50 μM is used for [AA] and 10 μM or 150 μM for [DAG]. Based on the preliminary studies, the maximum intracellular [AA] can reach 100 μM, though the cell membrane can disintegrate when [AA] is over 10 μM [12]. In simulations, [Ca$^{2+}$] is set between 0 μM and 10 μM, and [AA] and [DAG] between 0 μM and 150 μM.

3.1. Effects of stimuli

First, the sensitivity of the PKCa responses to changes in [Ca$^{2+}$] is examined. [AA] and [DAG] are kept constant: 50 μM for [AA] and 150 μM for [DAG]. Figure 2 presents example simulations in which sine waves of time period 10 s and 50 s are used for the Ca$^{2+}$ stimulus. The concentrations of all species follow the stimulus, and, in particular, sine waves induce oscillating PKC activity. The difference between the curves on the right and left in Figure 2 is that [Ca$^{2+}$] changes between 0 μM and 2 μM on the left and between 1 μM and 3 μM on the right. The changes in [PKCa] are not as great on the right than on the left because the stimulus is not reset to 0 μM on the right. In the case where the time period of the signal is 50 s and [Ca$^{2+}$] changes between 0 μM and 2 μM or between 1 μM and 3 μM, the amplitude of [PKCa] is 0.05 μM and 0.02 μM, respectively. Amplitude is given as the absolute value of one-half of the distance between local maximum and minimum concentrations. Even though the PKCa responses are only partially resembling sine waves, we use the term amplitude here as a measure to detect changes in [PKCa] when different types of stimuli are used. In the case where the time period of [Ca$^{2+}$] is 10 s and [Ca$^{2+}$] changes between 0 μM and 2 μM or between 1 μM and 3 μM, the amplitude of [PKCa] is 0.02 μM and 0.01 μM, respectively. It can be observed that the longer the time period of the Ca$^{2+}$ stimulus, the greater the amplitude of [PKCa]. If the Ca$^{2+}$ stimulus attains 0 μM during the simulation, the amplitude of [PKCa] is greater compared to the sine wave stimulus which does not attain 0 μM. When the Ca$^{2+}$ stimulus is a sine wave and the constant [AA] is increased from 50 μM to 150 μM, also the [PKCa] values are increased, and when the constant [AA] is decreased to 1 μM, the [PKCa] values are decreased.

![Figure 2. Effects of varying Ca$^{2+}$ stimulus. X1) Ca$^{2+}$ stimulus, X2) PKCa response to stimulus in X1, and X3) detailed presentation of X2.](image)

![Figure 3. Effects of varying Ca$^{2+}$, AA, and DAG stimuli. X1) Ca$^{2+}$, AA (-), and DAG (- -) stimuli, and X2) PKCa response to stimuli in X1.](image)
Next, the sensitivity of the PKC pathway responses to changes in all inputs is tested. When the Ca$^{2+}$ stimulus is a sine wave, and [DAG] and [AA] are triangular waves, [PKCa] has a sine wave form with the same time period as the Ca$^{2+}$ stimulus but it also follows the triangular forms of the AA and DAG stimuli. Figure 3 shows simulation results when different sine wave stimuli are used for all second messengers. If [Ca$^{2+}$], [AA], and [DAG] change between 0 μM and 2 μM, 45 μM and 55 μM, and 45 μM and 55 μM, respectively, the form of [PKCa] follows the form of [Ca$^{2+}$]. However, when [Ca$^{2+}$], [AA], and [DAG] change between 0 μM and 2 μM, 0 μM and 50 μM, and 0 μM and 150 μM, respectively, the form of [PKCa] follows all stimuli. In summary, the model output follows the Ca$^{2+}$ stimulus in most cases. However, we find that the greater the amplitudes of [AA] and [DAG], the clearer their effects on [PKCa].

### 3.2. Effects of parameter values and concentrations

After changing the parameter values of the model, the shape of the [PKCa] wave in time remains the same but the actual concentrations vary. We take a closer look at two reactions, R6 and R7, containing four parameters. We change one parameter value at a time. When the value of $k_1$ is changed by 20 % and 50 % greater, [PKCa] is increased by 0.02 μM and 0.04 μM, respectively. When the value of $k_{-1}$ is changed 20 % and 50 % greater, [PKCa] is decreased by 0.02 μM and 0.04 μM, respectively. If the parameter value is very small, the value is not really changed after increasing it. Therefore, [PKCa] is the same as the original [PKCa]. When all parameters are changed, [PKCa] remains the same as the original [PKCa]. It can be stated that the greater the value of $k_1$, the greater the [PKCa] response and also the greater the value of $k_{-1}$, the smaller the [PKCa] response.

The change of initial concentrations one at a time does not have much effect on [PKCa]. We change the initial [PKCi], [CaPKC], and [AADAGPKC$^*$] by factors of 10 smaller and greater. The initial value has to be at least 1 nM to have an effect on [PKCa]. In most cases, the shape of [PKCa] remains the same but the concentrations vary.

### 4. DISCUSSION

In this study, we employ a differential equation model of the PKC signal transduction pathway [5] and study by computational means the effects of stimuli, parameter values, and initial concentrations on the behavior of the model. The results show that the behavior of the model is more or less robust, and even significant changes in the parameter values (±50 %) and initial concentrations do not promote the model to enter into a different dynamic state. The behavior is mostly determined by the changes in intracellular Ca$^{2+}$, but also DAG and AA can have an effect on the dynamics, depending on their concentration. Further comparative studies, both experimental-based and computational, are needed to assess the suitability of the studied PKC model in the context of modeling the signaling in cerebellar granule neuron. However, the model will provide an interesting test case for the development of stochastic simulation models, as well as for the development of simulation-optimization procedures for estimating unknown values of model parameters. For these two research purposes, the model needs to be relatively simple in the sense that it should include less than 25 parameters and express robust behavior. Based on our studies the PKC model satisfies the criteria.

### 5. ACKNOWLEDGMENTS

This study was financially supported by the Academy of Finland, project nos. 79854, 80455, and 104508 to M.-L.L., and 213462 (Finnish Centre of Excellence Program 2006-2011) as well as the Ulla Tuominen Foundation and the Foundation of Technology to T.M.

### 6. REFERENCES


EXTRACTION OF QUANTITATIVE GENE EXPRESSION DATA FROM THE IMAGES OF GENE EXPRESSION PATTERNS WITH PROSTACK AND iSIMBioS

Anna Matveeva\(^1\), Konstantin Kozlov\(^1\) and Maria Samsonova\(^1\)

\(^1\)Department of Computational Biology, Center for Advanced Studies, St. Petersburg State Polytechnical University, 29 Polytechnicheskaya ul., St. Petersburg, 195251, Russia, mdespb@mail.ru, kozlov@spbcas.ru, samson@spbcas.ru

ABSTRACT
The development of multicellular organisms involves the differential expression of many genes. Thus knowledge about spatial and temporal patterns of gene expression is crucial for understanding development. Typically researchers analyze spatial patterns of gene expression in multicellular organisms by visual inspection of photographic images obtained in the \textit{in situ} hybridization experiments. A profound limitation of this approach is that it captures little of 2D and 3D information and provides only rough quantitative information. Thus the development of new algorithms and methods for processing of images to accurately quantify the level of gene expression at cellular resolution remains one of the important tasks for bioinformatics.

Here we describe a method to quantify gene expression levels per nucleus in the \textit{Drosophila} blastoderm embryos without loss of spatial information.

1. INTRODUCTION
One of the important problems of bioinformatics is the development of an integrated problem-solving environment that allows users to perform a variety of tasks related to scientific computing, including processing and analysis of digital images of biological objects acquired with light, electron or confocal microscope, data manipulation, and scientific visualization.

Efforts with different rate of success have been made in this direction for years by both corporations and communities. The most interesting examples are \textit{VisiQuest} from AccuSoft Corporation and \textit{SCIRUN}. These packages have their own advantages and disadvantages but unfortunately lack important problem specific features and are not easily extendable.

In this work, we present a new software package ProStack (\textit{Pro}\textit{cessing} \textit{Stacks}) integrated in information management system known as iSIMBioS (Integrated Service Infrastructure for Molecular Biology Systems).

ProStack is capable to process 2D and 3D digital images of biological objects. It includes all methods needed to extract quantitative information from experimental data, such as patterns of gene expression in early \textit{Drosophila} embryo. Quantitative information on gene expression is essential to discover dynamical mechanisms underlying gene regulation by mathematical modeling [1, 2].

We describe the method for building complex workflows to process the digital images of patterns of gene expression. This method is applied to images of segmentation gene expression obtained by confocal scanning microscopy [4]. The segmentation genes are involved in \textit{Drosophila} embryogenesis at the syncytial blastoderm stage. We also present preliminary results on development of automated method to control the quality of the whole processing procedure.

2. MATERIAL AND METHODS

2.1 ProStack and iSIMBioS

ProStack implements more than 40 operations that include domain specific methods to correctly orient the embryo images and to extract quantitative information from nuclei and cytoplasm. Domain-independent procedures include rotation and cropping, plain and adaptive thresholding, edge detection, non-linear contrast enhancement, image reconstruction, noise reduction, interpolation, distance and watershed transformations and pixel sub-sampling.

The processing methods are available for a user through command line interface \textit{prostak} and application program interface \textit{parus} that can be linked statically or dynamically. All processing methods are exported to the iSIMBioS information management system as workflow modules (W\textit{M}s) by using wrappers written in Perl programming language. In this system the workflow can be visually constructed by use of the graphical user interface Pegas (Fig. 1) that provides convenient environment for all groups of scientists ranging from the beginners to experts.
2.2 Images of the segmentation gene expression patterns

There are 4 classes of segmentation genes expressed at the syncytial blastoderm stage, namely maternal coordinate, gap, pair-rule and segment-polarity genes. This classification is based on the analysis of mutant phenotypes and patterns of gene expression. The majority of segmentation genes code proteins that act as transcription factors.

For several years we investigate the dynamical mechanisms controlling segmentation [1,2,4] and the important component of this work is the quantification of the gene expression levels.

We acquire quantitative data from the images of gene expression patterns obtained by confocal laser scanning microscopy of fixed embryos. The embryos are stained with fluorescence tagged antibodies to visualize gene expression patterns at the RNA or protein level [4]. Each embryo was stained for Even-skipped protein and the other gene product. Additionally, each embryo was stained with a nuclear marker in order to identify the nuclei location. Each fluorescent stain is detected in a single channel. For each channel three optical sections were scanned. As a result for each embryo nine 1024x1024 pixels grayscale images in TIFF format without compression were obtained.

The gain of the microscope photomultiplier is set for each channel by selecting an embryo exhibiting the spatial pattern characteristic of maximal expression and adjusting gain so that a few pixels are saturated. Offset for each channel is set by setting pixels away from the embryo equal to zero. This calibration is used for scanning all embryos on one slide, and has proved highly reproducible between slides.

In this work, to develop the quantification method we use the images of 16 embryos stained for Even-skipped protein, \( \text{lacZ} \) mRNA of the even-skipped promoter-reporter construct and histones. These images are used to extract quantitative information on expression of the native \textit{even-skipped} gene (by monitoring the level of Even-skipped protein), as well as on expression of the reporter \textit{lacZ} gene under control of the \textit{even-skipped} promoter. The histone proteins are used as nuclear marker.

2.3 Acquisition of quantitative data

Previously, the extraction of quantitative information on gene expression was performed with Khoros, now available as VisiQuest from AccuSoft Corp. [3, 4]. To develop in-house tool we apply Pegas and ProStack to construct three scenarios for segmentation of images of gene expression patterns and extraction of quantitative information.

![Figure 1. The screenshot of Pegas. The main window consists of two frames. The left one displays the list of available WM's on each server machine. The right bigger frame shows how to construct the workflow. We illustrate how to select the output port for the WM called as max.](image1)

![Figure 2. The images of Drosophila embryo stained for Even-skipped (Eve) (a), \( \text{lacZ} \) mRNA expressed under control of the even-skipped (eve) promoter (b), and histone proteins (c) after combining z-sections, rotation and cropping.](image2)
image in which only those pixels that cover the area occupied by the embryo are “on”. The mask is used to calculate the rotation angle and to remove spurious pixels at the embryo borders. At the next step the embryo images are rotated and cropped (Fig. 2).

The second Smooth Mask workflow is constructed to precisely find the area occupied by the embryo in the image. The smooth mask is a binary image in which this area is marked by the foreground pixels (Fig. 3a). A gain, three embryo images obtained after rotation and cropping are compared to construct the pixel maximum image and the histogram of this resulting image is equalized. Next several median filters with different structural elements are applied. To smooth the border of the mask we apply distance transform followed by thresholding, edge detection and filling.

The third workflow called as Segmentation serves to extract quantitative information. The nuclear mask is constructed by smoothing the histones image with local histogram and median filters followed by watershed transformation and edge detection. The outlines of nuclei in the histones image overlaid with nuclear mask are presented in Figure 3b. The coordinates of the nuclei centroids expressed as per cent of the embryo length and width are then calculated by averaging the pixel coordinates for each segmented nucleus in the mask. The average fluorescence intensities (relative gene expression levels) for Even-skipped protein and lacZ mRNA in segmented nuclei are obtained by averaging the pixel values within each nucleus of each image.

2.4. Estimation of the segmentation quality

Usually visual inspection is used to estimate the nuclear mask quality. However, as the number of segmented nuclei in an embryo is about 2,500, an automated error checking procedure is required.

We estimate the quality of nuclear mask from the nuclear channel (histones) image by considering two classes of pixels. The pixels of the first class come from the area occupied by a nucleus and outlined with nuclear mask, while the second class encompasses all the pixels outlined by the watershed domain with the exception of pixels that are “on” in the nuclear mask. To estimate the accuracy of segmentation we calculate the ratio of variances in pixel values between and inside these two classes. For correctly segmented image this measure should be larger than for the incorrectly segmented one and consequently can serve as an accuracy estimator.

3. RESULTS

The workflows have been applied to process the images of gene expression patterns. Figure 3c presents the result of quantification of eve expression in the representative image. For clarity only the nuclei from the central 10% strip are plotted.

To check the validity of the estimator we have compared its values for a nuclear mask obtained with the algorithm proposed here and after one and two additional erosions applied to the mask. These erosions destroy the shape of nuclei. It appeared that for the representative embryo the initial value of the estimator 1.37 decreases to 0.67 and 0.28 after the first and second erosions respectively. For the mask obtained with the VisiQuest package the corresponding values of the estimator are 1.11, 0.26 and 0.02. This result shows, that the estimator proposed here can be used to evaluate the segmentation quality, however further experiments are necessary to develop a robust method.

Figure 3. Summary of quantification of gene expression in a triple stained embryo. Drosophila blastoderm embryo at late cleavage cycle 14, immunostained for Even-skipped (Eve), lacZ mRNA of the even-skipped (eve) promoter-reporter construct and histones. (a) Smooth whole-embryo mask; (b) detail of an overlay of the outline of the nuclear mask with the original nuclear channel (histones) image; (c) quantified gene expression data for Eve. Vertical axis represents fluorescence intensity, horizontal axis, position along the A-P axis (where 0% is the anterior pole). Only values from nuclei located in the middle 10% (D-V) of the embryo are plotted.

No defects have been revealed by visual inspection of nuclear masks constructed from the rest of 16 images. For these images the mean value of the estimator is 1.10, standard deviation is 0.23, i.e. very close to the estimator value for the representative image.
4. DISCUSSION

We developed a new software package ProStack and applied it to quantify the levels of gene expression in the confocal images. While the conventional confocal microscope software is able to quantify the pixel intensities in the regions-of-interest, special algorithms are still required to find these regions, e.g. nuclei. Currently ProStack works on images with the resolution of 8 bits per pixel, the images of higher resolution can be successfully read, but their processing is not implemented.

As was reported in [3] each optical section is usually scanned several times and these scans are averaged by the confocal microscope software to reduce the experimental noise. Our recent results show that the averaging procedure does not remove the noise completely. We are working now to characterize the statistical properties of this noise in the experimental images in order to improve the segmentation quality.

5. ACKNOWLEDGEMENTS

The support of this work by NIH Grant RR07801, NWO-RFFI № 047.011.2004.013 grant and Federal Contract № 02.467.11.1005 is gratefully acknowledged.

6. REFERENCES


ANALYZING BOOLEAN NETWORK DYNAMICS USING ATTRACTOR BASIN STRUCTURE

Matti Nykter\textsuperscript{1,2}, Juha Kesseli\textsuperscript{1}, Ilya Shmulevich\textsuperscript{2}, and Olli Yli-Harja\textsuperscript{1}

\textsuperscript{1}Institute of Signal Processing, Tampere University of Technology, P.O. Box 553, FI-33101 Tampere, Finland
\textsuperscript{2}Institute for Systems Biology, Seattle, WA, USA
matti.nykter@tut.fi, juha.kesseli@tut.fi, is@ieee.org, olli.yli-harja@tut.fi

ABSTRACT

In this work, we analyze the state space properties of Boolean networks. We examine how different properties of the state space emerge with different network ensembles. We show that some of these properties can be used to characterize the dynamical behavior of a network, specifically whether the network operates in ordered, critical, or chaotic regime. In addition we introduce a new method for analyzing the attractor basin structure of a Boolean network. We also discuss how this method can be used to characterize dynamical behavior through the information content of the state space.

1. INTRODUCTION

The behavior and structure of a dynamical system, e.g. a network of biomolecules in a living cell, can be modeled using a variety of models. Modeling can be done at different levels of detail depending on the type of the model class used. One of the most studied coarse-scale model classes is the Boolean network model. This model class has been applied in many different fields and for different types of applications.

It is common to analyze such networks in terms of their dynamical behavior. The response of the model is measured under different types of perturbations and with different inputs. Based on the observed behavior conclusions about the dynamics can be made. There are also approaches for analytically formulating the dynamical properties of an ensemble of networks, such as the annealed (mean field) approximation [1].

An alternative approach for studying the dynamics of the system is to construct a complete description of its state space, that is to evaluate all the possible states and state transitions of the system. Currently this approach is applicable only for small networks. In this work we address the state space analysis using Boolean networks as prototypes of nonlinear dynamical systems.

The state space representation of a Boolean network shows all the trajectories and attractors that are present in the network. Therefore, state space analysis supersedes the dynamical analysis since all information about the dynamics is incorporated into the state space. Even though the state space includes a large amount of information, it is not obvious how this information can be utilized to understand the dynamical behavior of the system. Here we compare the applicability of different state space properties for characterizing dynamical properties of the network. In addition, we introduce new techniques to analyze the state space of a network.

The structure of the paper is as follows. In section 2 we give definitions of Boolean networks and of several state space properties and show how they characterize different ensembles of networks. In section 3 we introduce a new way to visualize the state space structure and show how this approach can be used to parameterize the dynamical regime. Finally, conclusions about the results are made in section 4.

2. STATE SPACE PROPERTIES

Let \( s_i \in \{0,1\}, i = 1,\ldots,N \), where \( N \) is the number of nodes in the network, be the state of node \( i \) of a Boolean network. The binary vector \( s = (s_1,\ldots,s_N) \) is then the state of the network. A transition from state \( s \) is defined as \( F(s) \) where \( F = (f_1,f_2,\ldots,f_N) \) and \( f_i : \{0,1\}^N \to \{0,1\} \) is a Boolean function of \( k_i \) effective variables. That is, the value of node \( i \) depends only on \( k_i < N \) inputs. In this model, all the nodes are updated synchronously.

Random Boolean networks can be obtained by choosing random functions \( f_i \), commonly parameterized by the so-called bias \( p = E[f_i(x)] \), the probability that the function outputs 1 on an arbitrary input vector \( x \). If \( p = 0.5 \), then the function is said to be unbiased. Similarly, the number of inputs \( k_i \) can be endowed with a probability distribution with mean \( K \). By suitably varying the parameters \( K \) and \( p \), a dynamical phase transition can be observed. The parameter

\[
\sigma = 2p(1-p)K
\]

(1)
determines the regime: chaotic for \( \sigma > 1 \) and ordered for \( \sigma < 1 \) [1, 2]. It is easy to see that for unbiased networks, the critical connectivity is \( K_c = 2 \).

A state space representation of a Boolean network for the purposes of this work includes all the possible states \( s \) and information about all the state transitions. That is, in the representation all the \( 2^N \) states of the state space.
$S = \{0, 1\}^N$ are stored with state transitions in a way that enables easy retrieval of attractor level information in addition to the immediate state transitions that can easily be computed as needed from the functions of the network. 

Next we define some common state space properties of a Boolean network $F$ [3]. A trajectory of length $n$ is a path from state $s \in S$ to state $t \in S$ over $n$ steps such that $F^{(n)}(s) = t$. Any trajectory must at some point arrive at a state that occurred earlier. This leads to a loop where the same states are repeated infinitely. This loop is known as an attractor cycle. The set of attractor states of a network is defined as

$$A_F = \{s|\exists n : F^{(n)}(s) = s\}. \quad (2)$$

Thus $|A_F|$ is the total size of attractors of the network. States that can not be reached from any other state are known as garden of eden states. By defining the set of these states as

$$G_F = \{s|\forall t : F(t) \neq s\} \quad (3)$$

we can denote the number of garden of eden states in the network as $|G_F|$. 

A trajectory from state $s$ to the attractor is called a transient. Transient length, that is, length of the shortest path from state $s$ to the attractor, is defined as

$$T_F(s) = \min\{k|F^{(k)}(s) \in A_F\}. \quad (4)$$

Size of the state transition can be parameterized as a self overlap $J$ [4], defined as the number of bits that are equal at two consecutive states, $J = N - |s \oplus F(s)|$. Here $\oplus$ is an XOR operator and thus $|s \oplus (s)|$ is the Hamming distance between states $s$ and $F(s)$.

A basin of attraction includes all transients from $G_F$ states that lead to the same attractor and can be defined using relation

$$R(s, t) = \begin{cases} 
1, & \exists k, l : F^{(k)}(s) = F^{(l)}(t) \\
0, & \text{otherwise},
\end{cases} \quad (5)$$

meaning that $s$ and $t$ are in the same basin if and only if $R(s, t) = 1$. It is easy to see that $R(s, t)$ is an equivalence relation, and thus, it can be used to partition the state space into $N_B$ basins unambiguously.

To study how different state space properties succeed in characterizing the dynamical behavior of a Boolean network we use specific network ensembles. We use random Boolean networks of size $N = 16$ with connectivity $K = 1, 2, 3, 4$ and bias $p = 0.5$ in this study. Unbiased networks with $K = 1$ are ordered, and thus, they do not separate different dynamical regimes reliably. Best parameters in terms of the amount of distribution overlap are $|A_F|$, $|G_F|$, and $T_F(s)$. Separation of different classes using these parameters is illustrated in Figure 1.

![Figure 1. State space properties of Boolean networks.](image1)

Number of basins $N_B$, attractor size $|A_F|$, transient lengths $T_F(s)$, number of garden of eden states $|G_F|$ and self overlap $J$ are shown for $100$ $K = 1, 2, 3, 4$ and $p = 0.5$ random Boolean networks with $N = 16$.

![Figure 2. Separation of different ensembles in feature space.](image2)

While the separation is evident there are some overlap between different ensembles.
Visual illustration of state space structure is usually done by drawing a tree that shows the basin structure of attractors [3]. Therefore, this illustration effectively shows the information that can be summarized by parameters $|A_F|$, $|G_F|$, and $T_F(s)$. A problem with this kind of an illustration is that it completely ignores the location and boundaries of different basins. In the next section we propose a new illustration technique that can be used to visualize how the basins are joined in the state space.

### 3. BASIN STRUCTURE

As the state space of a Boolean network includes $2^N$ states, it can be represented as a two dimensional grid with $2^\lceil N/2 \rceil$ vertical and $2^\lfloor N/2 \rfloor$ horizontal nodes. To show the basin structure of the network we denote each basin with an individual symbol. Thus we have a two dimensional grid (an image) where each basin is coded with a different symbol (color / shade). Examples of this kind of a representation are shown in Figure 3.

From this proposed illustration one can effectively see the basin structure of the networks. The patterns in the figures show the organization of the basin structure. Probability to change the attractor by a bit flip is on average smaller when the basin consists of large continuous areas or regular patterns. On the other hand, highly scattered and irregular basin structure indicates that small perturbations can easily perturb the state to other basins. Thus, this illustration offers an effective way to obtain insight into the robustness of the system. Ordered dynamics can be distinguished from chaotic systems using the observed irregularities. This kind of an insight is not directly observable from traditional illustrations.

As the number of different symbols in the illustration is equal to the number of attractors, it is fair to ask whether the number of attractors directly affects the illustration. In Figure 4 we have drawn illustrations from $K = 1, 2, 3, 4$ networks with two attractors. It can be seen that there is a clear correspondence in patterns to Figure 3. Thus, the basin structure is not affected by the number of attractors. In the trivial one attractor case, however, all figures will obviously appear the same.

As there is an evident difference between the basin structure of network ensembles, it is of interest to study whether the dynamic regime can be determined on the basis of the basin structure alone. Obviously, the randomness of the illustration determines how chaotic the underlying network is. As we want to know how random the basin structure is, that is how much information is embedded in the basins, it is natural to study their information content. This can be done for instance by studying how the information is scattered in the frequency domain or by using information theoretic criteria.

In the frequency domain we can determine the amount of information simply by looking at how many different frequencies are needed to represent the illustration. We use the Walsh transform [5] to obtain a frequency domain representation of the basin illustration. The number of non-zero transform coefficients is shown in Figure 5. While there is an overlap with the coefficient distributions due to the outliers, they seem to separate the ensembles well.

A well known information theoretic criterion for determining randomness is mutual information. It is an en-
tropy based similarity measure and thus can be used to compare the shared information content of two distributions

\[ I(X; Y) = H(X) + H(Y) - H(X, Y) \] (6)

where \( H(\cdot) \) denotes entropy. When mutual information is applied to basin structures, it proves out to be unable to separate different basins from each other. While the basin structures have similar patterns, they are not identical up to a change of variables or relabeling of the basins even for networks from the same ensemble. Thus, one basin can not be used to predict the structure and label of another. In terms of joint entropy this means that \( H(X, Y) \sim H(X) + H(Y) \). Thus mutual information between two basin structures is effectively zero for most kinds of basins.

Another measure of information is Kolmogorov complexity, which is defined as the minimum length of a computer program that is needed to construct a given object. Thus the main difference between mutual information and Kolmogorov complexity based distance measure is that while mutual information measures statistical distance between two distributions, Kolmogorov complexity measures distance between two individual objects. Recent developments in information theory have shown that Kolmogorov complexity based definition of the universal information distance (UID) can be applied to compare the information content of individual objects with remarkable success [6, 7]. However, as Kolmogorov complexity cannot be computed, UID can not be directly applied in practical applications. Remarkably, an approximation that asymptotically has the same universal properties as the UID has been proposed. This approximation is known as normalized compression distance (NCD) and here Kolmogorov complexity is approximated by a real life compression algorithm. Normalized compression distance is defined as

\[ d_{\text{NCD}}(x, y) = \frac{C(xy) - \min(C(x), C(y))}{\max(C(x), C(y))}, \] (7)

where \( C(x) \) is the compressed size of an object and \( C(xy) \) is compressed size of a concatenation of objects \( x \) and \( y \) [7].

We apply multidimensional scaling (MDS) to the distance matrix \( D_{\text{NCD}} \) that includes the NCD distances between the state spaces of all the generated random networks. The result is shown in Figure 6. It can be seen that while there are a number of outliers, different dynamical regimes can still be observed. It is also interesting to note that the critical regime shows the most variation, while ordered and chaotic regimes are more compactly clustered. This is consistent with the hypothesis that the networks in the critical regime are most evolvable, and thus exhibit the widest variety of behavior.

4. CONCLUSIONS

We have shown that some state space properties can be used to characterize the dynamic regime of the network. We propose a new illustration technique that can be used to gain insight into the basin structure of the network. We have seen that the regularities in this illustration are correlated with the dynamical behavior. Basin structure can thus be used to characterize the dynamical regime.

We have compared the information content of the basin structures and shown that both frequency domain and information theoretic approaches can be used to measure the properties of the basin structure. Kolmogorov complexity based approach has also provided an important insight about network dynamics, namely that the basin structures show the most variation in the critical regime, an indication that these networks are the most evolvable.

5. REFERENCES


USING REGULARIZED DYNAMIC CORRELATION TO INFER GENE DEPENDENCY NETWORKS FROM TIME-SERIES MICROARRAY DATA

Rainer Opgen-Rhein and Korbinian Strimmer

Department of Statistics, University of Munich, Ludwigstrasse 33, D-80539 Munich, Germany
opgen-rhein@stat.uni-muenchen.de, korbinian.strimmer@lmu.de

ABSTRACT

Graphical models allow to understand regulatory interactions among genes and gene products in a cell, and hence contribute to an enhanced understanding of systems biology. Here we investigate a graphical model that treats the observed gene expression over time as realizations of random curves. This approach is centered around a regularized dynamic correlation. Here we investigate a graphical model that treats the observed gene expression over time as realizations of random curves. This approach is centered around a regularized dynamic correlation and introduce a regularization technique for the "small n, large p" domain

2. METHODS

2.1. Setup and Notation

We consider data from a typical gene expression time course experiment. For p genes (variables) and n subjects (replications) mRNA concentrations are measured over a time interval \([A, B]\). This results in functional observations \(f_{ik}(t)\) where \(1 \leq i \leq n\) and \(1 \leq k, l \leq p\). We assume all functions \(f_{ik}(t)\) to be square-integrable so that the functional inner product

\[
\langle g(t), h(t) \rangle = \frac{1}{B-A} \int_A^B g(t)h(t)dt
\]

exists, where \(g(t)\) and \(h(t)\) are any of the observed functions. The time average of \(f_{ik}(t)\) may then be conveniently expressed by \(\langle f_{ik}(t) \rangle_1\). The average over the \(n\) replicates gives the empirical mean function \(f_k(t) = \frac{1}{n} \sum_{i=1}^n f_{ik}(t)\).

In practice, however, the functions \(f_{ik}(t)\) are not continuously measured but rather obtained by experiments at discrete time points \(t_j\), with \(1 \leq j \leq m\) and \(A = t_1 < t_2 < \ldots < t_{m-1} < t_m = B\). Note that the time points need not be equidistant. If one assumes a linear approximation of \(g(t)\) and \(h(t)\) the inner product of Eq. 1 turns into the weighted sum

\[
\langle g(t), h(t) \rangle \approx \sum_{j=1}^m g(t_j)h(t_j) \frac{\delta_j + \delta_{j+1}}{2(B-A)}
\]

where the \(\delta_j = t_j - t_{j-1}\) are the time differences between subsequent measurements (with \(\delta_1 = \delta_{m+1} = 0\)).

In the random effects representation of Dubin and Müller, 2005 [7] each observed \(f_{ik}(t)\) is a realization of the random function

\[
f_k(t) = \mu_k(t) + \epsilon_{ok} + \sum_{u=1}^{\infty} \epsilon_{uk}\eta_u(t),
\]

where \(\epsilon_{ok}\) and \(\epsilon_{uk}\) are random variables with \(E(\epsilon_{ok}) = 0\) and \(E(\epsilon_{uk}) = 0\). The fixed time dependent mean function with zero time average \((\mu_k(t), 1) = 0\), \(\mu_k + \epsilon_{ok}\) represents the static random part and the remaining terms describe the dynamic random part. In Eq. 3 the \(\eta_u(t)\)
are orthonormal basis functions with zero time average \( \langle \eta_k(t), 1 \rangle = 0 \).

In this notation the empirical mean function \( \hat{f}_k(t) \) is an estimate of \( E(f_k(t)) = \mu_k(t) + \mu_0 k. \) As \( \mu_k(t) \) has time average zero we are also able to identify the two components of \( E(f_k(t)) \) by using \( \hat{\mu}_k = \langle f_k(t), 1 \rangle \) and \( \hat{\mu}_0 = \hat{f}_k(t) - \hat{\mu}_0 k. \)

### 2.2. Dynamical Correlation

#### 2.2.1. Measuring similarity between two exactly known curves

Suppose for a moment that we have sufficient data to estimate the expression levels through time of two genes \( k \) and \( l \) exactly, i.e. that we know the mean functions \( E(f_k(t)) \) and \( E(f_l(t)). \) In order to understand the functional connection between these two variables a measure of similarity between the two curves is required. Dubin and Müller, 2005 [7] suggest to introduce the notion of dynamical correlation with the informal proposition that “if both trajectories tend to be mostly on the same side of their time average (a constant) then the dynamical correlation is positive; if the opposite occurs, then dynamical correlation is negative”.

This immediately leads to the following straightforward definition of dynamical correlation between two curves \( g(t) \) and \( h(t). \) First, calculate the time-centered functions \( g^C(t) = g(t) - \langle g(t), 1 \rangle \) and \( h^C(t) = h(t) - \langle h(t), 1 \rangle. \) Then define the variances as

\[
\text{Var}(g(t)) = \langle g^C(t), g^C(t) \rangle
\]

and

\[
\text{Var}(h(t)) = \langle h^C(t), h^C(t) \rangle.
\]

Finally, compute the standardized functions \( g^S(t) = g^C(t)/\sqrt{\text{Var}(g(t))} \) and \( h^S(t) = h^C(t)/\sqrt{\text{Var}(h(t))}, \) and obtain the correlation by

\[
\text{Cor}(g(t), h(t)) = \langle g^S(t), h^S(t) \rangle.
\]

#### 2.2.2. The general case including sampling error

The above definition of dynamical correlation for a single curve extends in a straightforward fashion to the case where each observed time course \( f_{ik} \) represents a noisy realization of the mean function \( E(f_k). \)

In order to estimate the correlation between two variables \( k \) and \( l \) first define the simultaneously time- and space-centered functions according to \( f_{ik}^C(t) = f_{ik}(t) - \langle f_{ik}(t), 1 \rangle. \) Note that here the inner product is computed over the mean function \( \bar{f}_{ik}(t). \) Based on the \( f_{ik}^C(t) \) the empirical estimate of the variance of variable \( k \) is then given by

\[
\hat{\text{Var}}_{ik} = \hat{\sigma}_{ik} = s_{ik} = \frac{1}{n} \sum_{i=1}^{n} \langle f_{ik}^C(t), f_{ik}^C(t) \rangle.
\] (4)

This allows to compute standardized residual functions \( f_{ik}^S(t) = f_{ik}^C(t)/\sqrt{s_{ik}} \) that form the basis for the estimate of dynamical correlation

\[
\hat{\text{Cor}}_{kl} = \hat{\rho}_{kl} = \hat{r}_{kl} = \frac{1}{n} \sum_{i=1}^{n} \langle f_{ik}^S(t), f_{il}^S(t) \rangle.
\] (5)

Correspondingly, the estimated dynamical covariance between variables \( k \) and \( l \) is simply

\[
\hat{\text{Cov}}_{kl} = \hat{\sigma}_{kl} = s_{kl} = \hat{r}_{kl} \sqrt{s_{kk} s_{ll}}.
\] (6)

This simple estimator of dynamical correlation exhibits several attractive properties. In particular, it is a generalization of the standard correlation for cross-sectional data. Specifically, if \( m = 1 \) and \( n > 1 \) then it reduces to the usual maximum-likelihood estimator of correlation. Furthermore, it is also applicable if there is only a single realization of each time series available \((n = 1, m > 1)\).

#### 2.2.3. Regularization

The above definition allows the inference of correlations between sets of curves. However, if there are only a few observations for a large number of variables (“small \( n \), large \( p \)”-problem), the unbiased empirical estimator is suboptimal in the sense that other, biased estimators may be constructed that are more efficient and exhibit higher accuracy in terms of MSE [8]. The pivotal element in successful learning of complex models from sparse data is regularization. It is possible to achieve a better estimation of dynamic correlation by means of shrinkage.

In the present case, we can construct a shrinkage estimate \( S^* \) of the dynamic covariance matrix by the convex combination \( S^* = \lambda S_{\text{Target}} + (1 - \lambda) S \) of the unregularized estimator \( S \) and a suitable target \( S_{\text{Target}}. \) The selection of the shrinkage parameter \( \lambda \) will have to take place in a data-driven fashion and has to meet some requirements. For instance, given a large sample size the shrinkage intensity \( \lambda \) must vanish. A simple rule to estimate the optimal shrinkage intensity can be found by minimizing the MSE risk function

\[
R(\lambda) = E \left( \sum_{k=1}^{p} \sum_{l=1}^{p} (s_{kl}^* - s_{kl})^2 \right).
\] (7)

It can be shown [9] that the minimum mean squared error \( R(\lambda^*) \) is achieved exactly and uniquely for the choice

\[
\lambda^* = \frac{1}{\sum_{k=1}^{p} \sum_{l=1}^{p} E \left( (s_{kl} - s_{kl}^{\text{Target}})^2 \right)} \cdot \left[ \sum_{k=1}^{p} \sum_{l=1}^{p} \text{Var}(s_{kl}) - \text{Cov}(s_{kl}, s_{kl}^{\text{Target}}) + \text{Bias}(s_{kl}) E(s_{kl} - s_{kl}^{\text{Target}}) \right].
\] (8)

Here we choose \( S_{\text{Target}} \) to be the diagonal matrix with the variances \( s_{kk} \) on the diagonal. Defining

\[
\bar{f}_{ik} = \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ik}^S(t_j) f_{il}^S(t_j) \frac{\delta_j + \delta_{j+1}}{2(B - A)n} u_{ij}
\] (9)
The unbiased empirical covariance equals
\[
\hat{\text{Cov}}(g(t), h(t)) = s_{kl} = \frac{1}{1 - \tau} \hat{f}_{kl} \tag{11}
\]
and find after some calculation the individual entries for
\[
\hat{\text{Var}}(s_{kl}) = \frac{\tau}{(1 - \tau)^2} \sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij} (f_{ijkl} - \hat{f}_{kl}^2) \tag{12}
\]
For scaling reasons \[9\] we apply shrinkage to the correlation matrix. The variances \(\text{Var}(r_{kl})\) of the empirical correlation coefficients can be estimated by applying the above formulae to the standardized data \((f_{ijkl}^0)\). This leads to the sample approximation of the shrinkage intensity
\[
\hat{\lambda}^* = \frac{\sum_{k \neq l} \hat{\text{Var}}(r_{kl})}{\sum_{k \neq l} r_{kl}^2}, \tag{13}
\]
which in turns allows to calculate the matrix of shrunken dynamical correlation coefficients.

2.3. Estimating Gene Association Networks Using Dynamical Correlation

The basic concept behind inferring a gene dependency network from the pairwise dynamical correlation is to investigate the correlation structure. However, we cannot simply use the correlations directly, because these represent only marginal dependencies and also include indirect interactions between two variables. Instead, we need to rely on the concept of partial correlation which describe the correlation between any two variables conditioned on all the other variables. It is straightforward to compute the matrix of partial dynamical correlations \(\hat{P} = (\hat{p}_{kl})\) from the correlation coefficients \(P = (p_{kl})\) via the inverse relationship
\[
\Omega = \hat{P}^{-1} = (\omega_{kl}) \tag{14}
\]
\[
\hat{p}_{kl} = - \frac{\omega_{kl}}{\sqrt{\omega_{kk} \omega_{ll}}} \tag{15}
\]
[10]. Applying these equations to estimates \(R = (r_{kl})\) of (dynamical) correlations allows to obtain estimates \(\hat{R} = (\hat{r}_{kl})\) of the associated partial (dynamical) correlations.

In order to test the significance of the correlations and to decide which of the possible edges to include in the resulting gene association network statistical tests are needed. In this paper we employ the “local fdr” network search \[1\], \[9\]. The false discovery rate (fdr) is the expected proportion of false positives among the proposed edges. The local fdr is an empirical Bayes estimator of the false discovery rate \[11\]. In the network search the local fdr is utilized to compute the posterior probability for an edge to be present or absent, and takes account of the multiplicity in the simultaneous testing of edges. The final network is obtained by visualizing all significant edges in an undirected graph.

3. RESULTS

We now employ shrinkage estimation of the (partial) dynamical correlation to a real world example and compare it with the results of the traditional GGM method. Specifically, we reanalyzed a microarray time series data set \[6\]. These data characterize the response of a human T-cell line (Jirkat) to a treatment with PMA and ioconomin, and consist of 10 time points with 44 replications each.

As approximation of the temporal expression of the 58 genes we used a linear spline and employed Eq. 2 for the functional inner product. After estimation of the dynamical correlations with Eq. 5 and regularization (section 2.2.3) we computed the associated partial correlation coefficients employing Eq. 14 and Eq. 15. Using the locfdr algorithm \[11\] we then identified significant edges. The resulting network is displayed in Fig. 1d.

For comparison we also compute the network as obtained by the classic GGM approach. For this analysis we ignored the dynamic aspects of the data and assumed that all measurements were taken at the same time point. Furthermore, we examine the influence of shrinking. This leads to the four networks displayed in Fig. 1.

Ignoring the time series aspects and using static correlation leads to less well-connected networks compared with the ones calculated by dynamical correlation. This indicates that our dynamical FDA-based estimator is able to extract additional information about the interaction among the investigated genes. Furthermore, shrinkage also improves the power of the network reconstruction. Hence, we conclude that the best of the four investigated methods to infer gene association networks is the one relying on regularized dynamical correlation.

4. CONCLUSION

A growing interest in genetics lies in observing and inferring the gene interactions over time. Here, we introduced a method to infer a regularized gene dependency network from functional data. It generalizes the static regularized GGM approach \[1\] and is able to unravel the dependency structure of longitudinal data across the whole time series. Furthermore, unlike many other time series methods FDA does not require equally spaced measurements. Note that in FDA unequal time points are accounted for by the weights employed in the functional inner product. Furthermore, our algorithm is easily implemented and computationally inexpensive. Shrinkage allows to improve the precision of the estimation and to extend the method to high dimensional data. In order to further develop our approach many extensions are conceivable. An important topic is the inclusion of auto-regressive aspects. While our method covers the dynamical correlation through time it is not able to account, e.g., for a time shift between any two variables. These dependencies and the associated time shifts could be accounted for by modeling the temporal mean via a system of differential equation.
5. ACKNOWLEDGMENTS

This work was supported by Deutsche Forschungsgemeinschaft (DFG) Emmy-Noether research award to K.S.

References


ABSTRACT

Many genes and proteins are required to carry out the processes of innate and adaptive immunity. We have identified and cataloged human immunology-related genes, which we call the immunome. The 554 identified genes and proteins were annotated and characterized. Disease relation was also taken into account. We identified numerous pseudogenes, many of which are expressed, and found two putative new genes. With this list of genes and proteins we intend to create a solid basis of further systems biological studies.

1. INTRODUCTION

The human immune system, which is one of the most complex biological machineries, has been widely investigated at the molecular, cellular and organ level in its normal state and during disease. Even with these extensive studies, however, we are still missing a comprehensive definition of this system at the molecular level. In the postgenomic era, research focus has shifted to complex systems, including the immune system [1]. For systems biology studies, it is necessary to have a clear and comprehensive definition of the immune system, including the genes and proteins that take part in immunological processes [2,3].

The identification of all immunological genes and proteins is difficult, because there are so many activities and processes involved in this system. One way to identify genes is to analyze immunity-related tissues and cells with microarrays. In a recent microarray study, 360 mouse genes were identified as being preferentially expressed in thymus, spleen, peripheral blood mononuclear cells, lymph nodes or in vitro-activated T cells [4]. Microarrays have also been widely used to analyze several immunological processes [5,6]. These studies have concentrated only on well-characterized processes, such as T cell activation or immunology-related diseases, and therefore cannot provide a comprehensive list of the genes of immunology. Immunology-related genes that are expressed in nonimmunological cells and tissues will also be missed with this type of analysis.

Another approach to gene identification is literature based: mining of immunology textbooks, publications and databases. The problem with literature mining is that it can lead to subjective selection. Gene ontology terms are useful systematic gene annotations in databases. However, their applicability to immunological gene identification is limited. Ontologies are not fully reliable or objective and are far from complete.

Pseudogenes are gene copies that are nonfunctional as a result of the disruption of their transcription or translation. Pseudogenes have several biological roles; for example, they may regulate gene expression and create a reservoir of diversity at a genetic or phenotypic level [7]. Recombination between the immunoglobulin Vh gene and its pseudogenes has been proposed to drive the formation of new genes [8]. Pseudogenes can also drive gene conversion to contribute to immunoglobulin heavy and light chain diversity [9]. An analysis of immunologically related genes would not be comprehensive without a detailed study of their pseudogenes.

Here we compiled a comprehensive list of genes and proteins of the human immunome for further systems biological studies. We identified and characterized 554 genes and proteins, together with their pseudogenes.

2. MATERIALS AND METHODS

2.1. Identification of immune system genes

Immunology-related genes and their corresponding protein sequences were collected from research articles, textbooks and electronic information sources. We concentrated on genes and proteins that are directly involved in immunological processes. In addition to clearly defined groups such as CD molecules, chemokines and their receptors, other essential genes were included. Genes were included when they are undoubtedly needed for immunity. Immunodeficiency-related genes were taken from ImmunoDeficiency Resource [10] and ID-bases [11]. Proteins that are expressed in nearly all cells were excluded. Only full-length genes were included; thus, the segments of immunoglobulins, B and T cell receptors and MHCs were excluded. In the case of signal-
ing molecules, only those involved in immunity-related cascades were included.

Altogether, 554 genes were obtained after rejecting close homologs. The systematic Entrez Gene IDs were used, because of inconsistencies in naming in other databases. Protein sequences were downloaded from NCBI GenBank. Chromosomal localization data were collected from our BioData database. Chromosome ideograms were drawn based on NCBI Homo sapiens Map View data (Human genome Build 35.1) and Mus musculus Map View data (Mouse genome Build 34.1). Genes related to diseases were collected from the BioData database.

2.2. Comparison of human and mouse immunomes

Mouse homologs of the genes were identified from the HomoloGene database, and human and mouse cDNA sequences were obtained from NCBI RefSeq. Perl scripts were written using some modules from the Bioperl Project for the comparison of mouse and human sequences at the cDNA and amino acid sequence levels. Synonymous mutations per synonymous sites (Ks) and nonsynonymous mutations per nonsynonymous sites (Kn) values [12] were calculated. The chromosomal localization data of the mouse orthologs in BioData were extracted from the genome builds.

2.3. Pseudogene analysis

Pseudogene relatives of the 554 genes were identified using a previously published approach [13] with some modifications. The identified pseudogenes were grouped into the following categories: known pseudogenes, pseudogene fragments, duplicated pseudogenes and interrupted processed pseudogenes. Blastn searches were performed against the human EST database at NCBI to find out whether the pseudogenes were transcribed. Sequences identified as new genes were further analyzed by the gene prediction programs GenScan [14] and Geneld [15]. Promoters were searched for the newly identified putative genes from a 1-kb upstream genomic DNA stretch using the PromoterInspector program [16].

3. RESULTS

3.1. Human immunome

We identified 554 genes that belong to the human immunome based on an exhaustive analysis of literature and databases. These genes and proteins represent many functions, such as cell surface recognition, transcription factors, DNA processing and adaptor proteins. According to the gene ontologies, 155 proteins are integral to the plasma membrane and 58 are extracellular. Of the proteins, 101 have receptor activity and 42 have chemokine activity.
Figure 3. Analysis of synonymous and nonsynonymous substitution rates for human and mouse immunome genes. (a) Number of orthologous human-mouse gene pairs in the various $K_a$ and $K_s$ value ranges. $K_a$ refers to nonsynonymous substitution rates per site (dashed line), whereas $K_s$ to synonymous substitution rates (continuous line). (b) Number of ortholog gene pairs in the various ranges of $K_a/K_s$ quotient ranges.

The distribution of the genes in the genome is not completely random. It was not surprising to us that none of the genes is localized on the Y chromosome, but these genes are also absent from the short arms of chromosomes 13, 14, 15, 18, 21 and 22. There are also regions with many immunome genes, such as 1q2 (CD genes), 1q13 (CD genes and killer cell immunoglobulin-like receptor [KIR] genes), 17q1 (chemokine ligand genes) and 2p13 (CD genes and lymphotoxin b receptor gene) and on the short arm of chromosome 19 (Figure 1). Some of these gene clusters are well known, such as the cluster of killer cell immunoglobulin-like receptor genes at locus 19q13.4 [17]. Disease-related genes also have a biased distribution. We found no immunodeficiency genes on chromosome 18 or Y, and we found a single gene on chromosome 8.

3.2. Comparison of human and mouse immunomes

We identified mouse orthologs for 399 genes and analyzed the colocalization of the ortholog pairs. Large stretches of the human genome are in corresponding positions in the mouse genome. Chromosomes 8 and Y lack immunology-related genes with mouse orthologs, the other human chromosomes have orthologs in 1 to 5 mouse chromosomes (Figure 2). Analysis of the sequence differences showed that synonymous nucleotide substitutions occur at an average rate of 0.65 substitutions per site (range: 0.26–1.59), whereas nonsynonymous substitutions occur at 0.18 substitutions per site (range 0–0.88; Figure 3a). $K_a/K_s$ quotient ranges are from 0.01 to 0.81 (Figure 3b).

3.3. Pseudogene analysis

We identified 3655 pseudogenes or fragments for 203 genes. The Pseudogene.org database lists pseudogenes for 84 immumome genes. The VPREB1 gene has the largest number of known pseudogenes (46), and almost half of them (22) have significant hits in the EST database. Most of the new pseudogenes were pseudogene fragments: 3188 fragments for 150 genes. The expression of the pseudogenes varied substantially based on EST data. The gene for interferon-γ receptor b chain (IFNAR2) had altogether 2114 fragments, but none of them had EST hits. The gene encoding CD163 had 259 fragments, 89 of which were expressed. Forty-nine genes had duplicated pseudogenes. We identified 16 interrupted processed pseudogenes for three genes. The gene for tyrosine 3/tryptophan 5–monooxygenase activation protein (YWHAZ) had seven duplicated pseudogenes, and the gene IFNAR2 had 12 interrupted processed pseudogenes.

We identified 282 putative genes as homologs for 59 genes. We further analyzed these results and accepted these putative genes only if the GenScan P value was $>0.75$ and the GenElD score was >0. We also required that the findings of the two gene prediction programs had to overlap on the same strand. Using these criteria, we identified 20 putative exons and two single-exon genes. The new single-exon genes are on chromosome 10 and are very close to each other. They have both been annotated as parts of the putative gene FLJ46361. We independently identified these genes as pseudogenes for three separate genes: CD5l, CD6 and CD163. We further identified three putative exons in two independent cases. Two of these exons are located close to the new genes on chromosome 10 (homologs of CD5L and CD6, CD6 and CD163, respectively), and the third one is on chromosome 19 (homolog of CEACAM1 and PSG1). Five putative exons that are homologous to CDH5 are located very close to each other (within a 12-kb stretch) on chromosome 5. These observations indicate extensive duplications in the near past. We identified the initial exon of a putative gene homolog of MIF along with a promoter sequence.

4. DISCUSSION

The human immunome consists of a group of diverse proteins. Both innate and adaptive immunity are dependent on many processes. This is reflected in the properties of the genes and proteins, which vary greatly in their function, gene ontology, subcellular localization,
expression, tissue distribution and protein domain content, among others.

A comparison of human – mouse ortholog gene pairs showed the speed of the evolution of the immunome after the divergence of the lineages. Substitution rates showed typical average values \cite{18}. The quotient, K_a/K_s, can be interpreted as the fraction of mutations that have been accepted or rejected by selective forces during a certain evolutionary period \cite{19}. During the 70 million years that separate the human and mouse genomes, only the most-conserved proteins and domains have remained unchanged. K_a values for 280 genes were \( \leq 0.2 \), meaning that less than 20% of the nucleotide point mutations cause amino acid changes (Figure 3a). By examining the distribution of K_a/K_s, two peaks can be detected at categories 0.05-0.1 and 0.2-0.25 (Figure 3b). For 90 ortholog pairs, K_a/K_s is <0.1. In these cases, more than 90% of the observed point mutations at the nucleotide level do not cause an amino acid change, because selective forces prevented most of the amino acid substitutions. This group is evolutionarily very conserved. About half of the proteins have K_a/K_s quotients that are <0.25, and the value is >0.5 for only 20 proteins.

The analysis of pseudogenes of the immunome genes provides an insight into the role of pseudogenes. By using EST data, we found that a number of pseudogenes are actually expressed. For example, for the cadherin-5 gene we identified five pseudogene fragments, all of which had EST hits with 100% identity throughout the length of the pseudogene.

5. CONCLUSION

A list of 554 genes and proteins has been compiled as basic elements of the human immune system. The members were characterized by genomic methods. This list intends to be a solid basis for further system biological studies of the human immune system.

6. REFERENCES

SELF-OVERLAP IN BOOLEAN NETWORKS

Pauli Rämö, Juha Kesseli and Olli Yli-Harja

Institute of Signal Processing, Tampere University of Technology,
P.O. Box 553, FI-33101 Tampere, Finland
pauli.ramo@tut.fi

ABSTRACT

Boolean networks are used to gain understanding on large, non-linear, and dynamical systems such as gene regulatory networks. The concept of self-overlap gives the probability that a network node keeps its value after one time step. We derive an iterative formula for the development of self-overlap based on the annealed approximation. We confirm the analytical results by performing numerical experiments. In addition, we study numerically some properties of typical attractors in Boolean networks. We find that the properties of attractors do not depend only on the network order parameters but also on other topological and dynamical characteristics of the network. Differences in the development of self-overlap give a possible explanation for the observed changes in the attractors.

1. INTRODUCTION

A Boolean network is a directed graph with \( N \) nodes. Nodes represent genes while graph arcs represent biochemical interactions between the genes. The in-degree of a node is the number of incoming connections to the node. \( p_k \) is the probability that a node has in-degree \( k \). Each node is assigned a binary output value and a Boolean function whose inputs are assigned according to the graph connections. Output value 1 represents an active gene while value 0 represents a non-active gene. The state of the network is the vector of node values at time \( t \). The network nodes are updated synchronously. Boolean function \( f \) associated with the node represents the biochemical interactions that regulate the gene in question. We denote the distribution of functions in the network by \( \mathcal{F} \). The in-degree and function distributions are connected to each other so that the degree distribution of the function distribution matches the network in-degree distribution.

In this work we study the concept of self-overlap in Boolean networks. Self-overlap \( s_t \) is the probability that a node keeps its value when the network is run for one time step. In other words, \( 1 - s_t \) is the normalized Hamming distance between consecutive states in the network.

2. ANALYSIS OF ANNEALED BOOLEAN NETWORKS

In our analytical treatment we use the annealed approximation. This means that we let network size \( N \) approach infinity and shuffle the network connections and functions after every time step. In addition, we assume that the network topology is random, i.e. the inputs to a node are chosen randomly among all the other nodes. These approximations allow us to use simple probabilistic methods to analyze the dynamical behavior of the network. We refer to finite size Boolean networks with fixed connections and functions as quenched Boolean networks. Unlike the trajectories in the annealed networks the ones in quenched networks finally reach an attractor.

Let us now consider what is the probability that a node has value 1 if we know the probability in the previous time step in the annealed network. The probability is given by bias-map

\[
g(b) = E_{f \in \mathcal{F}} \left[ \sum_{x \in \mathbb{B}^K} f(x) P(x|b) \right],
\]

where

\[
P(x|b) = b^{|x|} (1 - b)^{K - |x|}
\]

is the probability of input vector \( x \) when \( b \) is the probability that a function input has value 1. Here, \( K \) is the in-degree of function \( f \) that is drawn from a desired function distribution and \(|x|\) is the number of 1’s in vector \( x \). The bias-map can be iterated by

\[
b_{t+1} = g(b_t).
\]  

This mapping may have non-trivial fixed point solutions depending on the chosen function distribution [1, 2, 3]. We assume that all biologically realistic function distributions reach unique stable fixed point \( b^* \) [1]. For function distribution \( \mathcal{F} \) average sensitivity

\[
\lambda = E_{f \in \mathcal{F}} \left[ \sum_{i=1}^{K} \sum_{x \in \mathbb{B}^K} f(x) \oplus f(x \oplus e_i) P(x|b^*) \right]
\]

can be used as a network order parameter. Here, \( \oplus \) is exclusive or operation and \( e_i \) is a unit vector with value 1 at the \( i \)-th position. In other words, \( \lambda \) is the average number of nodes that are perturbed one time step after we have perturbed the value of a randomly chosen node. The initial state is chosen so that each node has probability \( b^* \) of being 1. We define that if \( \lambda < 1 \), the network is stable. If \( \lambda = 1 \), the network is critical, and if \( \lambda > 1 \), the network is chaotic. We define parameter \( \Delta \lambda \) as the derivative of the bias-map at the bias-map fixed point

\[
\Delta \lambda = g'(b^*).
\]
This parameter has been found to be important for example in determining the number and lengths of attractors in quenched Boolean networks [4].

Let us consider a situation where we have two annealed networks running in parallel [5]. Both networks have the same functions and connections at each time step. Each node pair may have four different value combinations (00, 01, 10, and 11). Probabilities \( p_{00}, p_{01}, p_{10}, \) and \( p_{11} \) for the combinations can be updated according to the following map

\[
p_{00}(t+1) = E_{x \in B} \left[ \sum_{x \in B} \sum_{y \in B} (1-f(x))(1-f(y))P(x,y,t) \right],
\]

\[
p_{01}(t+1) = E_{x \in B} \left[ \sum_{x \in B} \sum_{y \in B} f(x)f(y)P(x,y,t) \right],
\]

\[
p_{10}(t+1) = E_{x \in B} \left[ \sum_{x \in B} \sum_{y \in B} f(x)(1-f(y))P(x,y,t) \right],
\]

\[
p_{11}(t+1) = E_{x \in B} \left[ \sum_{x \in B} \sum_{y \in B} f(x)f(y)P(x,y,t) \right],
\]

where

\[
P(x,y,t) = p_{00}(t)(1-x)^T(1-y)p_{01}(t)(1-x)^Ty \times \ldots \times p_{10}(t)x^T(1-y)p_{11}(t)x^Ty.
\]

Naturally, the sum of these probabilities adds up to one \( p_{00} + p_{01} + p_{10} + p_{11} = 1 \). For this reason, the map has three degrees of freedom. We define \( b^1 \) as the probability that a node has value 1 in the first annealed network and \( b^2 \) as the probability that a node has value 1 in the second annealed network at time \( t \). \( p \) is the probability that a node pair has different values. We find that the connections between these probabilities are

\[
b^1 = p_{10} + p_{11},
\]
\[
b^2 = p_{01} + p_{11},
\]
\[
\rho = p_{10} + p_{01},
\]
\[
p_{00} = 1 - \frac{1}{2}(b^1 + b^2 + \rho),
\]
\[
p_{01} = \frac{1}{2}(b^2 - b^1 + \rho),
\]
\[
p_{10} = \frac{1}{2}(b^1 - b^2 + \rho),
\]
\[
p_{11} = \frac{1}{2}(b^1 + b^2 - \rho).
\]

Probability \( \rho \) that a node pair has different values at time \( t \) can be written as map

\[
\rho_{t+1} = h(\rho_t, b^1, b^2)
\]

\[
= E_{x \in B} \left[ \sum_{x \in B} \sum_{y \in B} f(x) \oplus f(y)P(x,y,t) \right],
\]

where \( b^1 \) and \( b^2 \) are iterated by using Eq. (1). This map can be used to determine how perturbations of magnitude \( \rho_t \) spread in the network. In particular, by choosing \( b^1_0 = b^2_0 = 1/2 \) we can study how perturbations to a random initial state propagate in the network.

We define self-overlap \( s_t \) as the probability that a node keeps its value when the (quenched) network is run for one time step [7]. The model with two parallel annealed networks can be used to estimate the development of self-overlap in the following way. Choose a random initial state. Its bias is on average 1/2. Run the network for one time step. The first overlap is on average 1/2 for every Boolean network. Now, choose the initial state and the state we have reached after one step. Set \( b^1_1 \) as the bias of the initial state (= 1/2) and \( b^2_1 \) as the bias of the second state and \( \rho_1 \) as one minus the first overlap (= 1/2). The second overlap is then \( s_2 = 1 - \rho_2 \) where \( \rho_2 \) is given by Eq. 2. In general, we get

\[
s_{t+1} = 1 - h(1 - s_t, b^1, b^2, \rho_{t-1})
\]

where \( s_1 = 1/2 \) and \( b_0 = 1/2 \). See Fig. 1 for a graphical illustration of the situation.

In the context of Boolean and gene regulatory networks, random functions have been of particular interest [6]. We define the distribution of random functions with single parameter \( p \). This is the probability that an entry in the truth table of a function has value 1. For random functions we have the result [5]

\[
s_{t+1} = 1 - 2p(1-p) \sum_{k=1}^{\infty} p_k(1 - s^k_t).
\]
This result has been reported in [7] for Boolean networks with a constant in-degree. Figure 2 shows the development of self-overlap for two kinds of random Boolean networks. Firstly, we have the development of self-overlap for two kinds of random Boolean networks with random functions ($p = 1/2$). Solid lines are for networks that have constant in-degree $K$ and dotted lines are for networks that have in-degree $2K$ with probability $1/2$ and in-degree $0$ with probability $1/2$.

Numerical experiments are needed to validate whether the presented model gives good estimates for self-overlap in quenched networks. First, we choose a Boolean network where every node has function $f = [01111110]$. This function yields non-trivial dynamical behavior for the network [1]. Second, we create a network of size $N = 500$ and run the network starting from a random initial state. Then, we calculate the normalized Hamming distances between the consecutive states. And finally, we repeat the experiment for 8000 times and average the results. The analytical estimate and the numerical results are presented in Fig. 3. We observe that the theoretical method gives a good estimate of the development of self-overlap in the network. We also perform tests with random functions and note that the numerical results follow closely the analytical results.

3. ATTRACTOR PROPERTIES

It seems plausible that we can use the concept of self-overlap in determining some state space and attractor properties of Boolean networks. However, connections between annealed and quenched dynamics in Boolean networks are not straightforward. First, we may hypothesize that if the self-overlap converges very rapidly to a high value, average transient times should be rather short and vice versa. Second, fixed point $s^*$ gives the final probability that a node remains the same after a single time step. Whether this holds approximately also for quenched Boolean networks is an interesting question. On the other hand, networks with $s^* > 0$ are chaotic and their numerical study can be difficult. Third, we can study closing probabilities that give the probability that an annealed trajectory intersects with itself. Bastolla et al. [8] have extensively studied this approach, but the results are mostly unsatisfactory. A more appropriate approach to examine attractors and other state space properties is to study the statistical properties of relevant nodes [4].

The full state space enumeration of large Boolean networks ($N > 100$) is virtually impossible. However, it has been found that many of the attractors in Boolean networks may have extremely small basins of attraction [4] or can be unstable [9]. Therefore, most of the attractors cannot be observed in practice. For these reasons, we choose to study the properties of typical attractors that can be found simply by running the network starting from a random initial state. We cannot find the number of typical attractors but we can easily study numerically their lengths and average transient times also for large Boolean networks. Transient time is the number of time steps that are needed to reach an attractor from a randomly chosen initial state. As an example, in Fig. 4 we have the distribution of typical attractor lengths and transient times for Boolean networks with random functions, constant in-degree $K = 2$, and network size $N = 40$.

As a simple test case we study typical attractors in networks with constant in-degree $K = 2$ and random func-
Figure 4. Lengths of attractors and transient times for typical attractors in Boolean networks with random functions, constant in-degree \( K = 2 \), and network size \( N = 40 \).

Figure 5. Attractor lengths (lines without dots) and transient times (lines with dots) of typical Boolean networks \( (N = 500) \) with a) random functions and constant in-degree \( K = 2 \) (solid lines) and b) random functions with in-degree \( K = 4 \) with probability \( 1/2 \) and in-degree 0 with probability \( 1/2 \) (dotted lines).

5. REFERENCES


PARAMETER ESTIMATION FOR HODGKIN-HUXLEY TYPE OF MODELS

Antti Saarinen*, Marja-Leena Linne and Olli Yli-Harja

Institute of Signal Processing, Tampere University of Technology, P. O. Box 553, FI-33101 Tampere, FINLAND, *antti.saarinen@cs.tut.fi

ABSTRACT
Our understanding of the input-output function of single cells has been substantially advanced by biophysically accurate multi-compartmental models. The large number of parameters needing hand-tuning in these models has, however, somewhat hampered their tractability and interpretability. In this paper we present the original Hodgkin-Huxley (H-H) gating model, introduce previously presented parameter estimation procedures for H-H type of models and discuss their applicability in stochastic modelling framework using the stochastic expansion of the H-H gating model and the stochastic one-compartmental multi-conductance model.

1. INTRODUCTION
Usually the tradeoff in parameter estimation for single neuron models is between realism and tractability. Typically, the more biophysical accuracy one tries to incorporate into the model, the harder the computational problem of parameter estimation becomes, as the number of nonlinearly interacting parameters increases.

The Hodgkin-Huxley (H-H) gating model [4] has been extensively employed over the last half a century to describe bioelectric phenomena related to normal and impaired electrophysiological functions. Since H-H gating model is relatively empirical, the associated modelling methodology requires estimating model parameters (including functions of membrane voltage) from experimental data. Until now, as is the case for most nonlinear models, parameter estimation has been carried out through nonlinear least-square fitting, which presents important limitations for the modelling methodology.

In practice, there are two modes for membrane current recording: one for analysis of ion channel populations (i.e., summed activity of a population of ion channels), and other for isolated ion channels. To date, the H-H gating model has been the most popular way to represent electrical activity of a ion channel population.

We continue by presenting the original H-H gating model and introducing previously developed estimation methods for it. We also introduce the stochastic expansion of the one-compartmental multi-conductance model for cerebellar granule cell [6] which is based on the H-H paradigm and discuss the applicability of presented parameter estimation methods in this context. Last, the problems in measuring similarity and estimating parameters in stochastic models are discussed.

The aim of this paper is to make a survey on different estimation methods presented for the deterministic and stochastic H-H type of models. The actual estimation of our stochastic neuron model will be the issue of another study.

2. METHODS AND RESULTS
2.1. Original Hodgkin-Huxley model
In the H-H gating model [4], a given ion channel is composed of several molecular components (gates). These change state under the influence of the electrostatic potential ($V_m$) existing across the cell membrane. Each molecular component has two states: open and closed. The channel is open when all of its molecular components are in the open state.

Precisely speaking, the variable $y_i$ represents the fraction of a population of gates in the open state, the kinetics of which obey

\[ \frac{dy_i(V_m,t)}{dt} = \alpha_i(V_m)(1 - y_i(V_m,t)) - \beta_i(V_m)y_i(V_m,t), \]  

(1)

where $\alpha_i(V_m)$ and $\beta_i(V_m)$ are, respectively, the forward and backward rates of transition between the open and closed states of the molecular gate $i$. We refer to the variables $y_i$ as the gating variables.

When in open state, the channel is seen as a resistive barrier to the passage of ions and has a fixed maximal conductance. The driving force for the passage of ion across the channel is the electrochemical gradient, which is given by $V_m - E_k$, where $E_k$ is the Nernst potential related to an ion species in question.

The membrane current for an ion species, denoted by $I_k(V_m,t)$, is

\[ I_k(V_m,t) = \bar{g}_k \left( \prod_{n=1}^{n_k} y_{n,k}(V_m,t)^{\lambda_{n,k}} \right) (V_m - E_k), \]  

(2)

where $\bar{g}_k$ is the maximal conductance, $n_k$ the number of distinct molecular gates of a channel type $k$, and $\lambda_{n,k}$ the number of similar components of a molecular gate $n$ in a channel type $k$.

The change in transmembrane potential is described using the ordinary differential equation

\[ C_m \frac{dV_m}{dt} = I_{app} - \sum_k I_k(V_m,t) - \frac{V_m - E_m}{R_m}, \]  

(3)
where \( R_m \) is membrane resistance, \( C_m \) membrane capacitance and \( I_{app} \) applied current.

### 2.2. Stochastic Hodgkin-Huxley type of model

There are numerous ways to incorporate stochasticity into the original H-H gating model, and there is a wealth of papers describing these expansions (see e.g. [2, 8]). In the previous stochastic models, stochasticity is incorporated by different means at the macroscopic and microscopic levels: stochasticity is incorporated into synaptic, conductance or voltage equations of the model or the behavior of discrete ion channels is modelled probabilistically to obtain the description of corresponding membrane conductance. Compartmental, multi-conductance models have mostly been used when stochasticity is incorporated into the synaptic input.

In this paper we consider the stochastic expansion of the one-compartmental multi-conductance model for cerebellar ganule cell [6]. In the model we have incorporated the stochasticity inherently present in the operation of ion channels into the gating variables. We describe the kinetics of a gating variable with the stochastic differential equation

\[
\frac{dW_i(t)}{dt} = \alpha_i(V_m)(1 - y_i(V_m, t)) - \beta_i(V_m)y_i(V_m, t) + \sigma dW, \tag{4}
\]

where \( \alpha_i(V_m) \) and \( \beta_i(V_m) \) are, respectively, the forward and backward rates of transition between the open and closed states of the molecular gate \( i \). In the equation \( W \) stands for Brownian motion, that is, a Gaussian stochastic process with independent increments. This means that all finite-dimensional distribution of \( W \) are Gaussian, \( W(0) = 0 \) almost surely, \( \mathbb{E}(W(t)) = 0 \) for all \( t \geq 0 \), and \( \text{Var}(W(t) - W(s)) = t - s \) for all \( t \geq s \geq 0 \).

The change in transmembrane potential is described similarly as in Eq. (3), and in addition we have included an auxiliary differential equation to describe the calcium dynamics.

Figure 1 illustrates the differences between the responses of the deterministic and stochastic neuron model. Based on our simulations, we can conclude that the stochastic neuron model reproduces behaviour seen in measurements from living neurons.

### 2.3. Parameter estimation for deterministic Hodgkin-Huxley type of models

Until now, parameter estimation has followed the practice introduced by Hodgkin and Huxley [4], where parameters are obtained from currents recorded under voltage-clamp conditions (voltage clamp data), using a high-resolution patch-clamp or voltage-clamp technique [3, 5]. In experimental studies, it has become a common practice to approximate the parameters associated with the steady-state functions of the model from currents obtained by varying the holding potential, and the time constants by fitting exponentials to the rising or decaying phase of currents. A mathematical analysis has shown that this approach may provide a good approximation to several model parameters if specific conditions are respected, and on the other hand can cause very large errors if not [1].

Briefly, cells are isolated, then a recording pipette is gently pressed against the membrane surface, and suction is applied to break the membrane under the pipette tip. The interior of the pipette is then in continuum with the intracellular space. An electronic feedback circuit allows one to simultaneously impose potential and record current on the electrode placed inside the pipette. The voltage-clamp stimulation consists of clamping the membrane potential at a holding potential \( V_H \) for an interval of time sufficiently long for ionic fluxes to reach steady-state. Then a step potential \( V_T \) is applied to allow a current to reach a new steady-state. The membrane current is recorded just after the application of the step until the steady-state.

Procedures currently used to estimate the parameters of the H-H gating model are based on nonlinear least-square fitting. It is well known that this approach has several drawbacks. Namely, the parameters obtained in this manner may correspond to a local minimum of the cost function undergoing minimisation. Thus not only may estimates lack accuracy, but there is no way to know whether other minima exist. Knowing the existence of such minima is important for the modeling methodology, because, considering the experimental error, several of them may correspond to plausible inverse solutions. In this case gating model predictions related to each minimum need to be examined. If model predictions are too wide-ranging,
then additional experiments need to be performed to obtain more specific current kinetics. While this is a highly desirable approach for the modeling of bioelectric phenomena, it is not possible to pursue in practice due to the inability to locate, in a rigorous manner, minima of the cost function.

The existence of local minima in the cost function to be optimised is particularly problematic. While the parameters are estimated from voltage-clamp data, in most applications, the model is employed in other conditions to simulate the generation and transmission of electrical impulses in biological tissues. Although parameter values associated with various local minima may allow good reproduction of the voltage-clamp data, they may produce different predictions when inserted in a more macroscopic model.

It should be noted that the original H-H model cannot be used for modelling structurally and functionally more complex neurons. More complex deterministic H-H type of models may contain several compartments and more parameters than the original H-H model. For estimating the full set of parameters in these complex models, only very few serious attempts have been presented [9]. Model development is usually done by hand-tuning.

During the past few years a couple of interesting methods have been presented to cope with the difficulties in least-square based estimation procedures for H-H gating models. In the following we point out two articles.

In the paper by Willms et al. [12], the authors consider whole-cell voltage-clamp data of isolated currents characterised by the H-H paradigm. They examine the errors associated with the typical parameter estimation method for the H-H gating model and show them to be unsatisfactorily large especially if the time constant of activation and inactivation are not sufficiently separated. They present an improved parameter estimation method which utilises all of the information in the voltage-clamp conductance data to estimate steady-state and kinetic properties simultaneously and illustrate its success compared to the standard method using simulated data.

In the work of Wang and Beaumont [10], the authors introduce an improved parameter estimation method for the H-H gating model. The authors provide a different approach to the estimation problem, which allows them to overcome all the limitations inherent to nonlinear fitting. Instead of fitting they invert the solution. Specifically, model parameters are obtained from multiple transformations applied to the solution, or equivalently, from an experimental data set. Such transformations enable one to deduce, for a given data point, disjoint ranges of parameter values which allow the model to exactly reproduce the solution. Using sufficiently large data sets and continuity criteria, it is possible to narrow down estimates to a specific value. Their main results are a more accurate estimation procedure and the ability to determine whether a data set sufficiently constrains the model.

2.4. Parameter estimation for stochastic Hodgkin-Huxley type of models

Despite the existing stochastic Hodgking-Huxley model, and other type of neuron models (for a review see, for example [2, 11]), the actual estimation of these models remains unexplored. The estimation of stochastic models is clearly a challenge as the existing deterministic estimation schemes are not fully able to utilise the information in stochastic responses.

When developing automatic parameter estimation methods for stochastic neuron models the approach has to be different from the procedures for deterministic models. Since the responses to repeated stimulation vary it is no longer sufficient to just consider the difference between the measured and simulated traces of voltage or current.

In a recent paper we have discussed different similarity measures for stochastic firing data [7]. In the work we propose the use of mean firing rate (i.e., the average frequency of action potential firing), mean of interspike intervals (i.e., the average amount of time between two consecutive action potentials), standard deviation of interspike intervals, coefficient of variation (i.e., the standard deviation of interspike intervals divided by the mean of interspike intervals), and histograms of interspike intervals in general in measuring the similarity between the experimental data and the model output. These similarity measures can be utilised when developing automated parameter estimation methods based on action potential firing data for stochastic neuronal models.

3. CONCLUSIONS AND FUTURE STUDIES

For future research the estimation of stochastic, biophysically accurate neuron models will be one of the key problems in the field of computational neuroscience. Since the existing realistic models are highly complex, it remains to be seen if case-specific solutions to the estimation problem need to be employed, or are we able to handle stochasticity on a general level. The stochastic neuron model developed by us serves as a good starting point for these kinds of studies.

In this paper, we have discussed the parameter estimation problem of the deterministic and stochastic Hodgkin-Huxley type of neuron models. Exploration of automatic parameter estimation methods for stochastic neuron models is imperative in order to develop realistic, biophysically accurate multi-compartmental models. The efficient and fast utilisation of data in the existing databases requires automated parameter estimation methods for neuronal models. Therefore it is crucial for the development of research in the neuroscience field to be able to consider automatic parameter estimation also in a stochastic context.

4. ACKNOWLEDGEMENTS

This study was financially supported by the Tampere Graduate School in Information Science and Engineering (TISE); the Academy of Finland, project No. 213462 (Finnish
Centre of Excellence program (2006 - 2011)) and project No. 106030.

5. REFERENCES


88
AUTOMATED ANALYSIS OF GOLGI APPARATUS DISPERSION IN NEURONAL CELL IMAGES

Jyrki Selinummi¹, Antti Lehmussola¹, Jertta-Riina Sarkanan², Jonna Nykky³, Tuula O. Jalonen³, and Olli Yli-Harja¹

¹Institute of Signal Processing, Tampere University of Technology, P.O.Box 553, FI-33101 Tampere, Finland
²Cell Research Center, Medical School, University of Tampere, FI-33014 Tampere, Finland
³Division of Biochemistry, Department of Biological and Environmental Science, Nanoscience Center, University of Jyväskylä, FI-40014 Jyväskylä, Finland
jyrki.selinummi@tut.fi, antti.lehmussola@tut.fi, riina.sarkanen@uta.fi

ABSTRACT
This paper presents an automated image analysis method for the quantification of subcellular structures. The analysis of fluorescently labeled cells is often used in biological studies for detecting intracellular structures and phenomena. Traditionally, the measurement and analysis has been performed only visually. To avoid main problems of the manual analysis, an automated image analysis based algorithm has been developed. We here analyze fluorescently labeled images of Golgi apparatus (GA) by dividing GA areas to each nuclei using clustering and quantify the dispersion of the fluorescent areas. Finally, we show that the dispersion correlates meaningfully with previous biological results.

1. INTRODUCTION
The analysis of fluorescently stained intracellular structures and properties is required in several biological studies. Manually the work, however, can be overly laborious and error prone. As presented in previous studies [1], digital image analysis offers tools for automation of the work, and quantitative and objective results that are not affected with human error. Automated image cytometry, the automated analysis of cell properties using microscopy images, is therefore becoming more popular. In this study we have implemented a method for the analysis of intracellular organelles that does not require user intervention after the imaging process.

Golgi apparatus (GA) has a central role as a protein processing, transport and sorting compartment of cells [2]. In neuronal cells, GA is involved in synapse development by providing storage of synaptic vesicle membrane proteins and sorting of proteins in precursor vesicles into developing synapses [3]. Golgi apparatus is known to undergo disassembly during some normal cellular processes such as mitosis [4], but also by several pharmacological agents [5]. However, these effects may be reversible [4, 5]. The recognition of normal organization of Golgi apparatus and especially differences in the normal structure, shape and localization is important when understanding intracellular traffic. Previously, it has been reported, that all-trans retinoic acid (RA) is known to cause Golgi dispersion by increasing the effect of certain immunotoxins [6, 7]. Also, it has been shown, that RA can change the membrane fluidity of intracellular organelles [8].

Cholesterol is a major component of cell membranes and membraneous organelles such as GA, and is known both to stabilize membrane structures as well as to provide flexibility [9]. The fragmentation of GA may also be related to neuronal degenerative diseases [10, 11, 12]. Differentiation experiments are commonly used to manipulate non-differentiated, usually cancer-derived cells, into more mature (in this study, normal neuron-like) state, and are important when studying neuronal development such as axon and dendrite formation and synaptic contact development.

We here wanted to analyze GA in neuronal cells when cells were exposed 10 days to all-trans retinoic acid (RA), cholesterol and their combination. We wanted to study if automated image analysis based approach could be applied to the analysis of Golgi apparatus structure and especially its dispersion. To the best of our knowledge, only few studies have been previously carried out about the automated analysis of Golgi apparatus dispersion [13]. We designed and implemented a k-means clustering [14] based image analysis algorithm, and tested the procedure with a small number of sample images consisting of fluorescently labeled nuclei and Golgi apparatus.

2. MATERIALS AND METHODS

2.1. Cell culture and differentiation treatment
SH-SY5Y neuroblastoma cell line (CRL-2266, ATCC, Manassas, VA, USA) was cultured in 1:1 Nutrient Mixture F-12K (all reagents from GIBCO, Invitrogen, Carlsbad, CA, USA, if not otherwise stated) and Minimum Essential Medium supplemented with 10% fetal bovine
serum (FBS), 1% antibiotic-antimycotic mixture, 2 mM L-glutamine and 1% non-essential amino acids (NEAA), in 5% CO2 humidified incubator at 37 °C. Medium was changed every three days. To induce differentiation cells were plated at 3500 cells/cm² at passage 35, cultured for 10 days and treated with 5 μM all-trans retinoic acid (RA, Sigma Aldrich, St. Louis, MO, USA), 10 μg/ml cholesterol (Sigma), 5 μM RA with 5 μg/ml cholesterol, or ethanol as control. Reagents were diluted in 96% ethanol so that the final ethanol concentration in cell culture never exceeded 0.1%.

### 2.2. Cell staining

For Golgi apparatus staining immunocytochemically, cells were fixed for 30 min with 4% paraformaldehyde (Sigma) in phosphate-buffered saline (PBS), washed three times in PBS and permeabilized for 15 min in 0.5% Triton X-100 (J.T.Baker, Phillipsburg, NJ, USA). Non-specific antibody binding sites were blocked with 10% bovine serum albumin (BSA, Sigma) and incubated for 1 h at room temperature with primary antibody (antigliantin, 1:1000, Biosite, Sweden) and 1 h at room temperature with secondary antibody (Alexa Fluor 488 goat antimouse IgG, 1:200 Molecular Probes, Invitrogen Corporation, Carlsbad, California, USA). Cell nuclei were stained for 30 min at room temperature with TOPRO-3 iodide nucleus stain (Molecular Probes, 1:100). Fluorescence was visualized and images captured with laser scanning confocal microscope (Zeiss Axiovert 100 M; Zeiss LSM 510, Jena, Germany), using several different magnifications.

### 2.3. Automated image analysis

The aim of the image processing procedure was to quantify the number of nuclei, the amount of Golgi apparatus, and to study the dispersion of the GA. The provided test images were in 24bit RGB format, consisting of 512×512×3 pixels. The first (red) channel included the nuclei, the second (green) channel consisted of stained GA, and the third (blue) channel was empty despite a 10 μm measure line that was present in each of the channels. The image was compressed using JPEG that caused some information to bleed between the channels. We implemented the algorithms using MATLAB 7.1 by The MathWorks, enhanced with the Image Processing Toolbox 5.1.

First we had to compensate for the different magnifications used. In each image, the measure line was extracted by thresholding each of the the channels (R,G,B), and performing a logical AND operation of the results. The images were then resized to the scale of 100 nm per pixel using nearest-neighbor interpolation. Second, the nuclei were extracted from the images. However, straightforward segmentation of nuclei was not possible, due to artifacts originating from staining and JPEG compression. Therefore, we used nonlinear filtering for reducing the effect of image artifacts in segmentation result. The filtering was performed with combination of rank filters [15] \( \zeta_{b,j} \) defined as

\[
\zeta_{b,j}(f)(s,t) = P_j\{f(s−x,t−y);(s−x),(t−y) \in D_f;(x,y) \in D_b\},
\]

(1)

where \( D_f \) is the domain of the image, \( D_b \) is the domain of the structuring element \( b \), and operator \( P_j \) returns \( j \)th percentile of the image inside structuring element. The images were filtered with the following combination of rank filters:

\[
o_{b,p}(f) = \zeta_{b,100−p}[\zeta_{b,p}(f)].
\]

(2)

A special case of the filter \( p = 0 \) is equivalent to grayscale morphological opening: A filter often used for removing bright objects in dark background, and in principle suitable for our task. In practice, however, morphological opening emphasized the shape of structuring element excessively, resulting as errors in the segmentation results. For small values of \( p \) for the filter \( o_{b,p} \) we were able to obtain similar properties for artifact removal with smoother and segmentation friendly output. In this study \( p = 5 \) and we used a 15 × 15 structuring element.

Subsequent to filtering, we segmented the nuclei images with Otsu’s thresholding [16]. Due to inefficient staining, the segmented nuclei contained some holes, which were filled using mathematical morphology. Finally, the overlapping nuclei were separated using watershed transformation [15]. The thresholding was then performed similarly to the GA image. After the binarization, we applied median filtering to the GA image in a 7x7 pixel window to remove false pixels (impulse noise) from the image. According to our previous knowledge, these isolated pixels are not parts of the GA, but are caused, for example, by staining residue.

The GA images also include GA pixels that are not part of any cell shown in the image but belong to the cells outside the image, and nuclei whose GA are located outside the image area. These extra GA areas and nuclei are removed using distance measures: All GA pixels further than 15 μm from the closest nucleus are removed, and all nuclei further than 5 μm from the nearest GA pixel are removed. These distance thresholds are also based on previous experiences in manual analysis of cells, and must be calibrated according to the cell types analyzed.

In order to enable cell-by-cell based analysis, we distributed the pixels consisting of GA to each \( k \) nuclei separately. This was done by clustering the coordinates of GA pixels into \( k \) groups using \( k \)-means clustering algorithm. The center of mass of each nucleus was selected as the initial cluster centers.

After the clustering, we acquired an estimate of GA dispersion for each GA cluster by calculating the median distance \( d_i \) of GA pixels \( x \) and \( y \) to the corresponding GA center of mass \( c_i \):

\[
d_i = \text{median}(\sqrt{(x−c_{ix})^2 + (y−c_{iy})^2}), \quad x, y \in GA_i
\]

(3)
A

Figure 1. From original image to quantized result. (A) Original image of nuclei and golgi. Illustrated in gray-scale for printing purposes. (B) Result image. Note the nuclei and golgi on the left side of the original image that have been discarded in the result image due to distance thresholds.

B

1. RESULTS

A proof-of-principle test of the algorithm was performed by analyzing dispersion of Golgi apparatus in neuronal images. Specifically we wanted to test whether there are any differences in the organelle structure or spatial organization of GA between selected differentiation treatments. For this study we acquired 20 images from which we selected the images that included four or more cells. This resulted in three images of control samples (ethanol), six images of cholesterol treated cells, five images of RA treated cells and four images with the combination of RA and cholesterol.

Visual observation of the biological results shows that when compared to control and cholesterol treated cells, some fragmented and dispersed Golgi apparatus can be seen in both RA and RA+cholesterol treated cells. This phenomena can also be observed from the automatically acquired results presented in Figure 2. The Figure 2 presents the median of dispersion $d_i$ over all nuclei of GA in each of the images. In our test image set, the dispersion is larger on average after RA treatment (squares) than in the control and cholesterol treated samples (circles). The automated analysis of one image takes approximately 15 seconds using a computer running MATLAB 7.1, equipped with 1.5GHz Pentium M ULV processor and 1.5GB RAM.

2. CONCLUSIONS

We presented here an automated image analysis method for the quantification of intracellular structures. Using automated image analysis, several problems of manual cytometry can be overcome: The results are perfectly reproducible and objective, the processing can be done automatically without laborious manual visual analysis, and quantitative results can be obtained easily. Main drawback of our current analysis approach is the fairly large number of user adjustable parameters. Although the pa-

\[ GA_i = \{ \text{pixels belonging to golgi apparatus } i \}, \]

where $c_{ix}$ and $c_{iy}$ are the row and column coordinates of the center of mass of $i$th GA, respectively. An example image is presented in Figure 1.

\[ d_i = \text{median dispersion of } GA_i \]

Visual observation of the biological results shows that when compared to control and cholesterol treated cells, some fragmented and dispersed Golgi apparatus can be seen in both RA and RA+cholesterol treated cells. This phenomena can also be observed from the automatically acquired results presented in Figure 2. The Figure 2 presents the median of dispersion $d_i$ over all nuclei of GA in each of the images. In our test image set, the dispersion is larger on average after RA treatment (squares) than in the control and cholesterol treated samples (circles). The automated analysis of one image takes approximately 15 seconds using a computer running MATLAB 7.1, equipped with 1.5GHz Pentium M ULV processor and 1.5GB RAM.

4. CONCLUSIONS

We presented here an automated image analysis method for the quantification of intracellular structures. Using automated image analysis, several problems of manual cytometry can be overcome: The results are perfectly reproducible and objective, the processing can be done automatically without laborious manual visual analysis, and quantitative results can be obtained easily. Main drawback of our current analysis approach is the fairly large number of user adjustable parameters. Although the pa-
rameters give the user possibilities to tune the algorithm to suit different cell segmentation needs, they can also delay the start of practical utilization of the algorithm.

In conclusion, we were able to identify that RA treatment causes alteration in the normal Golgi apparatus structure in neuronal cells. RA treatment could be separated from the other images by analyzing the median dispersion of Golgi in the input images, and despite the small number of sample images the results suggests that biologically meaningful information can be obtained, although no reliable conclusions can be drawn due to the low number of sample images. Future research will include quantification of other subcellular structures, e.g., lysosomes, and we will also test the algorithm for a much larger image data set to enable statistical significance testing of the results.

5. ACKNOWLEDGMENTS

The work was supported by funding from Biomaterial and Tissue Engineering Graduate School to JRS, by the Academy of Finland to JN (Contract # 102161), partly supported by the Academy of Finland to JS, and OYH, and partly by EU project BioPattern No. 508803 to JS.

6. REFERENCES


ON THE DISTRIBUTION OF SMALL AVALANCHEs IN RANDOM BOOLEAN NETWORKS

Roberto Serra¹, Marco Villani¹, Alex Graudenzi¹, Stuart A. Kauffman²

¹DSSC, Università di Modena e Reggio Emilia, via Allegri 9, I-42100 Reggio Emilia
²Institute for Biocomplexity and Informatics, University of Calgary
2500 University Drive NW, Calgary AB T2N 1N4, Canada
rserra@unimo.it, mvillani@unimo.it, graudenzi.alex.19435@unimo.it, skauffman@ucalgary.ca

ABSTRACT

The distribution of small avalanches of gene perturbations is analytically studied in the quenched model of random Boolean networks, providing formulae which hold at a very good approximation for networks where the number of connections per node is much smaller than the number of nodes. The expressions are particularly simple and elegant in the case of a Poissonian distribution of outgoing connectivities. Comparisons with simulations of a large network, with 6000 nodes, show very good agreement.

1. INTRODUCTION

Among the models of genetic regulatory networks, random Boolean networks (briefly, RBN) are well known and, in spite of drastic simplifications, they have proven useful to uncover generic features of the behaviour which emerges out of the interaction of very many genes in a cell [1].

In previous works [2,3] we have shown that RBN can be used to model the perturbations which occur in gene expression levels in cells of the yeast S. Cerevisiae which are subject to the knock-out (i.e., permanent silencing) of a single gene. Moreover, the simplest non trivial Boolean models already provide a satisfactory approximation to the distribution of avalanches.

To be precise, the perturbation we refer to is the model analogous of a single gene knock-out: at a certain time point, the state of one of the nodes is permanently clamped to the value 0. The evolution of the unperturbed network is compared to that of the perturbed one; a gene is said to be affected (or perturbed) if its value differs in the two networks in at least one time step. The avalanche corresponding to a given knock-out is the set of perturbed genes (including the one which has been knocked-out). Therefore the size of an avalanche in the model cannot be smaller than 1, nor larger than the number N of nodes.

The original study was based on simulations, which are very heavy for networks comprising more than 6000 nodes: therefore an extensive exploration of the effects of the different parameters has not been possible.

An approximate formula for the size distributions of avalanches in the annealed model of RBN has been introduced by Ramo et al [4], and provides a good approximation to some results obtained in simulations also with the quenched model, for avalanches comprising more than 5-10 genes. Recall the in the quenched model all the connections and Boolean function of a given network are fixed, while they continuously change in different time steps in the annealed model. In this paper we present an analytical method to determine, with a very good approximation, the size distributions of small avalanches in quenched random Boolean networks. Since the quenched model has a more direct physical meaning in the case of genetic networks (where the circuit is fixed) results obtained in this way can be more easily interpreted (and the limitations due to the approximations better appreciated).

Let us consider a network composed of N genes, or nodes, which can take either the value 0 (inactive) or 1 (active). In a classical RBN each node has the same number of incoming connections k_in and its k_in input nodes are chosen at random with uniform probability among the remaining N-1 nodes (multiple connections from the same node being prohibited). It then turns out that the distribution of outgoing connections per node follows a Poisson distribution:

\[ p_{out}(k) = e^{-A} \frac{A^k}{k!} \]  \hspace{1cm} (1)

where, since every connections must have both ends, \( A = <k_{out}> = k_{in} \).

The output (i.e. the new value of a node) corresponding to each set of values of the input nodes is determined by a Boolean function, which is associated to
that node, and which is also chosen at random, according to some probability distribution; the simplest choice is that of a uniform distribution among all the possible Boolean functions of $k_q$ arguments. However, a careful analysis of some known real biological control circuits [5] has shown that there is a strong bias in favour of the so-called canalyzing functions. A Boolean function is said to be canalyzing if at least one value of one of its input nodes can uniquely determine its output, no matter what the other input values are.

Both the topology and the Boolean function associated to each gene do not change in time (we refer here of course to the so-called quenched model, [6]). The network dynamics are discrete and synchronous.

In order to analyze the properties of an ensemble of random Boolean networks, different networks are synthesized and their dynamical properties are examined. The ensembles differ mainly in the choice of the number of nodes $N$, the input connectivity per node $k$, and the choice of the set of allowed Boolean functions.

2. THE DISTRIBUTION OF AVALANCHES

Let us compare the evolution of two identical RBN networks, which start from two initial conditions which differ for one bit only: in particular, one of the networks will be considered as unperturbed (“wild type”, briefly WT). The initial condition for the perturbed (“knocked-out” or KO) network differs from the previous one because one of the “1”s has switched to “0”. We will refer to this node as the root of the perturbation. We will also assume that the root of the perturbation is clamped to the value 0 in the KO network.

In order to simplify the writing, if there is a link from node A to node B we will call A an “input node” of B and B an “output node” of A. We will also call “nodes of level 1” the output nodes of the root, “nodes of level 2” the output nodes of those of level 1, etc. Since the network may have cycles, this tree-like terminology is ambiguous, in the sense that one node might be attributed to different levels. For the sake of definiteness, let us then define the level of a node as the smallest of its levels, computed as described above. We will see below that in the case of small avalanches the analysis of perturbations can indeed be reduced to that of a tree.

So, at time $t=0$ the Hamming distance between the two networks is 1 (note that $t=0$ is the time when the perturbation, ie the knock-out, is performed). At time 1 only a few other nodes may have been affected, namely those which are output nodes of the root node. However, it may happen that some, or all, of these nodes remain in the same state as that of the WT network, even if one of their inputs has changed. We may now compute the probability that at time 1 no node other than the root differs in the two networks: in this case no further changes may take place at later stages, and the size of the avalanche will be one.

Let then $p_{out}(k)$ be the probability that a node has $k$ outgoing connections, and let $q_{\leq 1}$ be defined as follows. For a node chosen at random, suppose that one (and only one) of its inputs is changed; then $q$ is the probability that this node does not change its value. If all the Boolean functions are allowed, $q=1/2$; if the set of Boolean functions is restricted, or if a bias is introduced, $q$ may take different values. We suppose here that

a. the probability that a node remains unchanged is the same whether an input which is 1 is changed to 0 as in the opposite case: $q_{1\rightarrow 0} = q_{0\rightarrow 1}$.

b. $q$, defined as above, is the same irrespective of the number of input connections $q(k)=q(k')$, for every $k,k'$

While it is possible to construct sets of Boolean functions which do not have these properties, they hold for all the sets which were used in our previous works as well as in most papers on RBN. Properties (a) and (b) hold if the set of Boolean functions is composed of canalyzing functions only, or of functions which are generated by attributing to each possible input a value chosen at random with bias $b$.

Let $p_1$ be the probability that an avalanche size is 1; $p_1$ is then the probability that all the output nodes of the root do not change; if there are $k$ outgoing connections, this probability is $q^k$; therefore, summing over the outgoing distribution:

$$p_1 = \sum_{k=0}^{N-1} p_{out}(k)q^k$$  \hspace{1cm} (2)

Let us now consider avalanches of size 2. We will limit ourselves to the case of large networks with a few connections per node (more precisely, the number of connections per node is much smaller than N-1); therefore the probability that an output node of the root is also one of its input nodes is negligible. In this case the probability that an avalanche size is 2 equals the probability that only one of the output nodes of the root (ie a node at level 1) changes its value, and that the perturbation does not propagate downwards to level 2 (ie that nodes which receive connections from the affected node do not change their value at the following time step).

Therefore

$$p_2 = \sum_{k=0}^{N-1} kp_{out}(k)q^{k-1} (1-q) \sum_{m=0}^{N-1} p_{out}(m) q^m$$

where $(1-q)q^{k-1}$ is the probability that only a particular node at level 1 changes its value, and the multiplication times $k$ comes from the fact that each one of the $k$ nodes at level 1 may be the one which is modified. We will assume that the probability of having very many outgoing connections (ie a number of outgoing connections close to the number of nodes) is very small: this holds in the case, which will be considered below, of a Poisson distribution, but it holds also in the important case of scale-free distributions [7]. Therefore, the summation in the
last term of the r.h.s. of the above equation may run up to N-1, and we recognize that it equals p₁, so

\[ p_2 = p_1 (1 - q) \sum_{k=0}^{N-1} p_{\text{out}}(k) k q^{k-1} \quad (3) \]

Let us now consider the probability p₃ to find an avalanche of size 3; it can be directly checked that it is the sum of two terms: one describes the case where the perturbation affects a node at level 1 and a node at level 2 (and no nodes at levels below the second), while the second term describes the case where two nodes at level 1 (and no node at level 2) are different in the two networks.

Here again we ignore possible feedbacks from perturbed nodes al levels 2 and 1 on the root, and from the node at level 2 on that at level 1. This is perfectly reasonable in cases where there are thousands of nodes (as in S. cerevisiae) and the number of connections per node is much smaller than the number of nodes. Therefore, by letting the sum run up to N-1 as in Eq. 3

\[ p_3 = p_3α + p_3β \]

\[ p_{3α} = (1 - q)^2 p_1 \left( \sum_{k=0}^{N-1} kp_{\text{out}}(k) q^{k-1} \right)^2 \quad (4) \]

\[ p_{3β} = \left[ p_1 (1 - q) \right]^2 \sum_{k=0}^{N-1} k (k - 1) \frac{1}{2} p_{\text{out}}(k) q^{k-2} (1 - q)^2 \]

The simplification associated to ignoring the feedbacks from perturbed nodes at a given level to nodes which had been previously perturbed at higher levels is justified as long as the avalanches are small with respect to the total size of the network. Whenever this approximation holds, the analysis is that of the branches of a tree, and it can continue along the lines given above.

Let us introduce the function

\[ F = \sum_{m=0}^{N-1} q^m p_{\text{out}}(m) \quad (5) \]

F is a function of q, N, and of the parameters which describe the distribution p_{\text{out}}. Since

\[ \frac{\partial F}{\partial q} = \sum_{k=0}^{N-1} p_{\text{out}}(k) k q^{k-1} \]

\[ \frac{\partial^2 F}{\partial q^2} = \sum_{k=0}^{N-1} k (k - 1) p_{\text{out}}(k) q^{k-2} \quad (6) \]

it is possible to rewrite Eq. 4 as follows

\[ p_1 = F; \]

\[ p_2 = (1 - q) F \frac{\partial F}{\partial q} \]

\[ p_3 = (1 - q)^2 F \left[ \left( \frac{\partial F}{\partial q} \right)^2 + \frac{1}{2} F \frac{\partial^2 F}{\partial q^2} \right] \quad (7) \]

It would be possible to continue the analysis, although no recursion formula has been found which gives the value pₙ for arbitrary avalanche formula sizes n. In any case, the validity of the approximation of ignoring feedback connections from lower to upper level nodes, as well as multiple connections from perturbed nodes to a single node, would break as the size of the avalanche increases; therefore the validity of such a formula would be limited. However, the distribution of the size of small avalanches can be accurately described.

Let us consider the case of a Poisson distribution of outgoing connections per node (see Eq. 1); in this case Eq. 5 becomes

\[ F = \sum_{k=0}^{N-1} q^k e^{-A} \frac{A^k}{k!} \approx \sum_{k=0}^{m} q^k e^{-A} \frac{A^k}{k!} = e^{-\lambda} e^{A} \]

therefore, introducing the variable \( \lambda = \ln(1/F) \):

\[ F = e^{-\lambda} \]

\[ \lambda = (1 - q) A \quad (8) \]

Note that \( \lambda = (1 - q) A \) equals the expected number of nodes of level 1 which modify their state (it is the product of the average number of outgoing connections times the probability that an output node of the root changes its value). It therefore coincides with the so-called Derrida exponent which has been often used to characterize the dynamics of RBN.

The above formulae (Eq. 7) for the probabilities of small avalanches become, in case of a Poisson distribution

\[ p_1 = e^{-\lambda} \]

\[ p_2 = \lambda e^{-2\lambda} \]

\[ p_3 = \frac{3}{2} \lambda^2 e^{-3\lambda} \quad (9) \]

It is easy to show that the general expression for the probability of an avalanche of size n is

\[ p_n = B_n \lambda^{n-1} e^{-n\lambda} \quad (10) \]

where the coefficients Bₙ can be computed in a straightforward albeit tedious way and the distribution of small avalanches can be explicitly given. Note that \( B_{n+1} \geq B_n \).
A point which is worth stressing is that, under the above assumptions, the avalanche size distribution is determined by the parameter $q$. This is likely to be the main reason why even unrealistic models with a fixed number of input connections per node may provide an adequate description of the avalanche distribution observed in biological networks [2].

We report here for completeness also the result for avalanches of size 4 and 5

\[
p_4 = \frac{13}{6} \lambda^3 e^{-4 \lambda}
\]

\[
p_5 = \frac{29}{8} \lambda^4 e^{-5 \lambda}
\]

A comparison between the results of the simulation of a RBN with more than 6000 genes ($k_{in}=2$) and the above formulae (see Figure 1) shows a very good agreement in the case where all the Boolean functions are allowed ($q=1/2$). Similar results have been obtained in the case where canalyzing functions only are allowed (by a direct calculation it can be shown that in this case $q=4/7$) and in the case where non canalyzing functions as well as the NULL function are excluded ($q=7/13$). The simulations were performed as described in [2].

![Figure 1. Comparison between the distribution of small avalanches given by eq. 10 and 12 and the average of simulated RBN of 6000 genes, with $k_{in}=2$ (q=1/2, $\lambda=1$)](image)

3. CONCLUSIONS

These results are simple and elegant, and they provide a very good approximation for small avalanches in large networks. The main approximation is related to neglecting the possible connections upwards and multiple connections from perturbed nodes to the same node, which are very reasonable hypotheses for avalanches of a few tens of genes in a network composed by thousands of nodes, provided that the number of connections per node is much smaller than the total number of nodes. These conditions seem to be satisfied by the S.cerevisiae genetic network as well as by those of other biological cells.

Note also that all the formulae of section 3, up to and including eq. 7, as well as the fact that the distribution of avalanches depends only upon the distribution of outgoing connections, hold for any distribution of outgoing connections, including of course those which are scale-free, while the validity of the simple and elegant equations 9-11 is limited to the Poisson case.

An interesting research line could be that of searching for the analogous of these formulae in the scale-free case.

The results described here have been used to describe the distribution of small avalanches in S. cerevisiae [Serra, Villani, Graudenzi & Kauffman, in preparation]; by considering the effects of the threshold used to binarize continuous microarray data, it has been shown that formulae 9-11 provide a very good description of avalanche size distributions for thresholds equal to or larger than 6.

Other interesting developments might involve the separate study of avalanches of up-regulated (and down-regulated) genes, and the analysis of susceptibilities (also defined in [1]).

4. REFERENCES

[1] S.A. Kauffman, The origins of order, Oxford Univer-
sity Press (1993)


man, “Perturbation in genetic regulatory net-

tion avalanches and criticality in gene regulatory networks”, submitted (2005)

man, “A model of transcriptional regulatory net-

