Conferences in Research and Practice in Information Technology

Volume 73

Intelligent Systems for Bioinformatics 2006

ACS

Australian
Computer
Society

Intelligent Systems for Bioinformatics 2006

Proceedings of the
2006 Workshop on Intelligent Systems for Bioinformatics
(WISB 2006),
Hobart, Australia, 4 December 2006

Mikael Bodén and Timothy L. Bailey, Eds.

Volume 73 in the Conferences in Research and Practice in Information Technology Series.
Published by the Australian Computer Society Inc.

Published in association with the ACM Digital Library.

Proceedings of the 2006 Workshop on Intelligent Systems for Bioinformatics (WISB 2006), Hobart, Australia, 4 December 2006

Conferences in Research and Practice in Information Technology, Volume 73.

Editors:
Mikael Bodén
School of Information Technology and Electrical Engineering,
University of Queensland,
Brisbane, Queensland 4072
Australia.
E-mail: mikael@itee.uq.edu.au

Timothy L. Bailey
Institute for Molecular Bioscience
University of Queensland
Brisbane, Queensland 4072
Australia
E-mail: t.bailey@imb.uq.edu.au

Series Editors:
Vladimir Estivill-Castro, Griffith University, Queensland
John F. Roddick, Flinders University, South Australia
Simeon Simoff, University of Technology, Sydney, NSW
crpit@infoeng.flinders.edu.au

Publisher: Australian Computer Society Inc.
PO Box Q534, QVB Post Office
Sydney 1230
New South Wales
Australia.

Conferences in Research and Practice in Information Technology, Volume 73.
ISSN 1445-1336.
ISBN 1-920-68254-6.

Printed, November 2006 by Flinders Press, PO Box 2100, Bedford Park, SA 5042, South Australia.
Cover Design by Modern Planet Design, (08) 8340 1361.

The *Conferences in Research and Practice in Information Technology* series aims to disseminate the results of peer-reviewed research in all areas of Information Technology. Further details can be found at http://crpit.com/.

Table of Contents

Proceedings of the 2006 Workshop on Intelligent Systems for Bioinformatics (WISB 2006), Hobart, Australia, 4 December 2006

Preface vii

Major Sponsors viii

Contributed Papers

Learning Causal Networks from Microarray Data 3
Nasir Ahsan, Michael Bain, John Potter, Bruno Gaëta, Mark Temple and Ian Dawes

Bacterial Promoter Modeling and Prediction for *E. Coli* and *B. Subtilis* with Beagle 9
Stefan R. Maetschke, Michael W. Towsey and James M. Hogan

Pathway to Functional Studies: Pipeline Linking Phylogenetic Footprinting and Transcription-Factor Binding Analysis 15
Nagesh Chakka and Jill E. Gready

Modified Global K-Means Algorithm for Clustering in Gene Expression Data Sets 23
Adil M. Bagirov, Karim Mardaneh

Clustering Replicated Microarray Data via Mixtures of Random Effects Models for Various Covariance Structures 29
S.K. Ng, G.J. McLachlan, R.W. Bean and S.-W. Ng

A Maximally Diversified Multiple Decision Tree Algorithm for Microarray Data Classification 35
Hong Hu, Jiuyong Li, Hua Wang, Grant Daggard and Mingren Shi

Comparing SVM Sequence Kernels: A Subcellular Localization Theme 39
Lynne Davis, John Hawkins, Stefan Maetschke and Mikael Bodén

Higher Order HMMs for Localization Prediction of Transmembrane Proteins 49
Stefan Maetschke, Mikael Bodén and Marcus Gallagher

Multi-stage Redundancy Reduction: Effective Utilisation of Small Protein Datasets 55
John Hawkins and Mikael Bodén

Linear Predictive Coding and its Decision Logic for Early Prediction of Major Adverse Cardiac Events using Mass Spectrometry Data 61
Tuan D. Pham, Honghui Wang, Xiaobo Zhou, Dominik Beck, Miriam Brandl, Gerard Hoehn, Joseph Azok, Marie-Luise Brennan, Stanley L. Hazen, King Li, and Stephen T.C. Wong

Automated Sub-Cellular Phenotype Classification: An Introduction and Recent Results 67
N. Hamilton, R. Pantelic, K. Hanson, J.L. Fink, S. Karunaratne and R.D. Teasdale

Quantification of Neural Images using Grey Difference 73
Donggang Yu, Tuan D. Pham, Hong Yan and Denis I. Crane

Author Index 81

Preface

Accurate and efficient computational tools are essential in order for biologists to make sense of the vast amounts of data being generated by high-throughput technologies such as genome sequencing and nucleotide micro-arrays. Existing intelligent systems offer powerful methods by which many biological questions can be addressed, ranging from the analysis of genomic and proteomic data, to the extraction of knowledge from biomedical text and imagery, to the modelling of biological processes and molecules. With large amounts of knowledge still waiting to be extracted from, for example, genomic data and biomedical text, and with new technologies continually creating novel data types, the field of intelligent systems in bioinformatics is receiving prime attention.

This book contains the proceedings of the first Workshop on Intelligent Systems for Bioinformatics, held 4 December 2006 in Hobart, Tasmania, Australia. The workshop was organised in conjunction with the Australian Joint Conference on Artificial Intelligence. The papers in this collection bring together work aiming to apply intelligent systems technologies to bioinformatics problems. Mathematical, probabilistic and computational methods (in the broad realm of intelligent systems) are applied in bioinformatics and computational biology, and important biological results that are obtained from the use of these methods. Contributions report on fundamental methodological research, on experimental and implementation issues involved in complex computations, and/or on the application of methods and programs that lead to discoveries of biological significance.

The papers presented here describe work on modelling biological processes, on clustering of numeric data, on classification and prediction of biological features and on image analysis. The specific biological biological applications include modelling of the gene regulatory cycle, clustering of gene expression data, predicting the subcellular localization of proteins, predicting adverse cardiac risk, and detecting neural differences in mice using image data. The researchers have applied a wide variety of algorithms and data representation languages, some novel, including causal networks, phylogenetic footprinting, the global K-means algorithm, expectation maximization, support vector machines, decision trees, and linear predictive coding. We envisage that researchers in artificial intelligence with an interest in scientific discovery, and bioinformatics researchers looking for the right tools, will gain from reading this collection of papers. It showcases significant and representative efforts that advance bioinformatics and computational biology.

The program committee of the workshop was chaired by Mark A. Ragan at the Institute of Molecular Bioscience, The University of Queensland, Australia and the Australian Research Council Centre for Bioinformatics. The program committee consisted of

- Adil Bagirov, The University of Ballarat, Australia
- Timothy L. Bailey, University of Queensland, Australia
- Regina Berretta, The University of Newcastle, Australia
- Mikael Bodén, University of Queensland, Australia
- Sarah Boyd, Monash University, Australia
- Vladimir Brusic, Harvard University, Australia
- Phoebe Chen, Deakin University, Australia
- Martin Frith, The University of Queensland, Australia and RIKEN, Japan
- Nicholas Hamilton, The University of Queensland, Australia
- Jim Hogan, Queensland University of Technology, Australia
- Lars Jermiin, The University of Sydney, Australia
- Geoff McLachlan, The University of Queensland, Australia
- Tuan Pham, James Cook University, Australia
- Alex Smola, National ICT Australia/The Australian National University, Australia
- Terry Speed, The Walter & Eliza Hall Institute, Australia
- Michael Towsey, Queensland University of Technology, Australia

The program committee reviewed all submitted contributions and at least two reviews (but usually three) were completed for each paper. The success of the workshop was due not only to the authors, but also to the advice from the program committee.

Mikael Bodén and Timothy L. Bailey

Brisbane, October 2006.

Major Sponsors

The committee would like to thank the ARC Centre for Bioinformatics and the ARC Centre for Complex Systems for their financial support.

ARC CENTRE FOR
COMPLEX SYSTEMS

Australian Government

Australian Research Council

ARC Centre in Bioinformatics

Contributed Papers

Learning Causal Networks from Microarray Data

Nasir Ahsan[†] **Michael Bain**[†*] **John Potter**[†] **Bruno Gaëta**[†‡]
Mark Temple[‡] **Ian Dawes**[‡]

[†] School of Computer Science and Engineering
[‡] School of Biotechnology and Biomolecular Sciences
University of New South Wales,
Sydney, Australia 2052
[*] Email: mike@cse.unsw.edu.au

Abstract

We report on a new approach to modelling and identifying dependencies within a gene regulatory cycle. In particular, we aim to learn the structure of a causal network from gene expression microarray data. We model causality in two ways: by using conditional dependence assumptions to model the independence of different causes on a common effect; and by relying on time delays between cause and effect. Networks therefore incorporate both probabilistic and temporal aspects of regulation. We are thus able to deal with cyclic dependencies amongst genes, which is not possible in standard Bayesian networks. However, our model is kept deliberately simple to make it amenable for learning from microarray data, which typically contains a small number of samples for a large number of genes. We have developed a learning algorithm for this model which was implemented and experimentally validated against simulated data and on yeast cell cycle microarray time series data sets.

1 Introduction

With the complete genomic sequences of increasing numbers of organisms available there are unprecedented opportunities for the biological analysis of complex cellular processes. In this new era of cell biology there have been major advances in the ability to collect data on a genome-wide scale by the use of high-throughput technology such as gene expression microarrays. However this data requires analysis beyond the level of individual genes; we need to investigate networks of genes acting within highly regulated systems.

Owing to the complexity of the systems to be studied a wide range of methods to model networks and learn them from data have been studied (Endy & Brent 2001, de Jong 2002). However, in our work we are not aiming to model the full dynamical systems of the cell, but rather to learn key causal features from data.

In this paper we investigate the use of causal networks to model relations between genes as measured in microarray data. More specifically, we explore learning the structure of a genetic regulatory network in this setting. Our approach to causal networks is based on a form of dynamic Bayesian network in which causality is represented in two ways: by relying on conditional independence assumptions to model the independence of different causes on a common effect; and by relying on time delays between cause and effect. By incorporating time delays into our model, we are able to deal with cyclic dependencies amongst genes. From a high-level perspective, we address the following problems:

1. How can we formalize the model of a causal network which combines probabilistic and temporal aspects, in particular, conditional independence and temporal precedence?

2. How can we learn such a model from observations of the variables over time in a robust and computationally efficient manner?

The paper is organised as follows. In Section 2 we introduce our representation for causal models, and in Section 3 we develop a learning algorithm for such models. In Section 4 we present results from experimental application of the approach to cell cycle time series microarray data.

2 A new probabilistic model

We introduce a probabilistic model that satisfies certain assumptions and addresses the problems of complexity discussed above. We then extend this probabilistic model to include time, and develop methods for estimating our probabilistic model from time series data, specifically microarray data. Note that in this paper our models contain only discrete random variables.

2.1 The $\mathscr{F}$-model

In the Bayesian network model an effect is conditionally independent of its ancestors given its immediate causes. This allows one to model an effect θ with immediate causes $\beta_1, \ldots, \beta_n$ as shown on the left in Figure 1. Unfortunately modelling an effect in this way requires the estimation of at least 2^n parameters, where n is the number of causes. Clearly the number of probabilities to be estimated increases exponentially in n.

However, if certain combinations of causes are known *a priori* to be unlikely they may be safely ignored in the interests of simplifying the problem. This suggests a way to reduce the number of distinct events that need to be modelled, effectively compressing the space of events for which probabilities need to be estimated. To do this we will use a function $\mathscr{F}$, called a *compression function*, to compress the joint probability distribution. In the standard Bayesian network framework the conditional probability for the dependence of an effect θ on its causes $\beta_1, \ldots, \beta_n$ is $P(\theta \mid \beta_1, \ldots, \beta_n)$. Using the compression function this becomes $P(\theta \mid \mathscr{F}(\beta_1, \ldots, \beta_n))$, as shown on the right in Figure 1.

If we assume that $P(\theta \mid \beta_1, \ldots, \beta_n) \equiv P(\theta \mid \mathscr{F}(\beta_1, \ldots, \beta_n))$, for some arbitrary $\mathscr{F}$, a joint probability distribution may be decomposed using $\mathscr{F}$-based conditional independence factors.

Definition 1 ***($\mathscr{F}$-model)*** *Let M denote a probability model over the random variable space $\Theta =$*

Copyright ©2006, Australian Computer Society, Inc. This paper appeared at The 2006 Workshop on Intelligent Systems for Bioinformatics (WISB2006), Hobart, Australia. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 73. Mikael Bodén and Timothy L. Bailey, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

$\{\theta_1,\dots,\theta_n\}$, *and let* $\mathscr{F}$ *be some arbitrary mapping over observations of* Θ. *An* $\mathscr{F}$*-model is the pair* $\langle\Theta, G\rangle$, *where* G *is a directed acyclic graph and each edge in* G *represents a dependence between a pair of variables, letting* $\alpha \equiv \beta_1,\dots,\beta_n$, *such that the set of immediate parents* $\alpha_i \subset \Theta$ *of* $\theta_i \in \Theta$ *in* G *render it independent of its other ancestors in* G. *The* $\mathscr{F}$*-model defines the following factorization of* M*:*

$$P(\Theta) = \prod_{i=1}^{n} P(\theta_i \mid \mathscr{F}(\alpha_i))$$

where $P(\theta_i \mid \mathscr{F}(\alpha_i)) \equiv P(\theta_i \mid \alpha_i)$.

Effectively, for a set of random variables $\{\theta_i,\dots,\theta_j\}$, $\mathscr{F}$ induces a new random variable $\mathscr{F}(\theta_i,\dots,\theta_j)$. Note that if our assumption above is valid with respect to the system under enquiry then the $\mathscr{F}$-model, by definition, is equivalent to a Bayesian network, assuming no cycles in the ancestor relation.

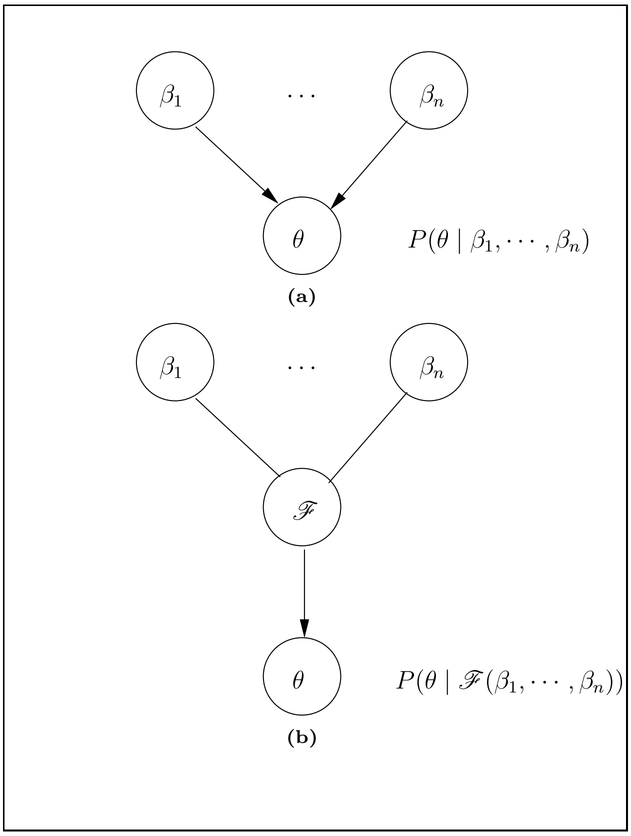

β_1
$\cdots$
β_n
θ
$P(\theta \mid \beta_1,\cdots,\beta_n)$
(a)
β_1
$\cdots$
β_n
$\mathscr{F}$
θ
$P(\theta \mid \mathscr{F}(\beta_1,\cdots,\beta_n))$
(b)

Figure 1: Modelling via (a) Bayes net dependencies and (b) $\mathscr{F}$ based dependencies.

2.2 Temporal extension of the $\mathscr{F}$-model

In order to model a causal system we must account for time. Therefore we provide a temporal generalization of the $\mathscr{F}$-model that accounts for temporal precedence and contiguity in time (Hume 1999). We extend the $\mathscr{F}$-model, by replacing the set of random variables Θ with a corresponding random process $\Theta(t)$ and its history at any time.

Definition 2 ***(Discrete Time Random Process and History)*** $\Theta(t) = \{\theta_1(t),\dots,\theta_n(t)\}$ *is a* discrete time random process *for* $t \in \mathbb{Z}$. *At time* t *the* history *of the random process is* $\mathscr{H}(t) = \bigcup\{\theta_i(\tau) \mid i \in 1,\dots,n, \tau < t\}$.

A causal model for $\Theta(t)$ requires the parent set $\alpha(t)$ for some $\theta(t)$ to be drawn from the history $\mathscr{H}(t)$. Each random process $\theta_i(t)$ has a finite set of parents $\beta_{i_1},\dots,\beta_{i_m} \subseteq \Theta$ with associated time shifts $\delta_{i_1},\dots,\delta_{i_m}$. Thus we may write the parent set $\alpha_i(t) = \{\beta_{i_1}(t-\delta_{i_1}),\dots,\beta_{i_m}(t-\delta_{i_m})\} \subseteq \mathscr{H}(t)$, allowing us to model causal relationships by the conditional probabilities $P(\theta_i(t) \mid \alpha_i(t))$.

We also make a *stationarity assumption*, i.e., we assume that the underlying causal relationships do not change over time. This is clearly not the case for cellular systems, but acts as a useful simplifying assumption. It follows that the parent sets $\alpha(t) = \langle\beta_1(t),\dots,\beta_m(t)\rangle$ and associated delays δ_i are independent of time t.

Hence the $\mathscr{F}$-model represents a stationary distribution by a graph where nodes are variables in Θ and edges are labelled with time shifts. Note that this graph may include cycles. However, since time shifts are positive, these cycles do not represent a cyclic probabilistic dependency, but merely that $\theta(t)$ depends on $\theta(t-\lambda)$, where λ is the period of a single cycle. Note that cycles are not permitted in the standard Bayesian network formalism which is restricted to acyclic graphs.

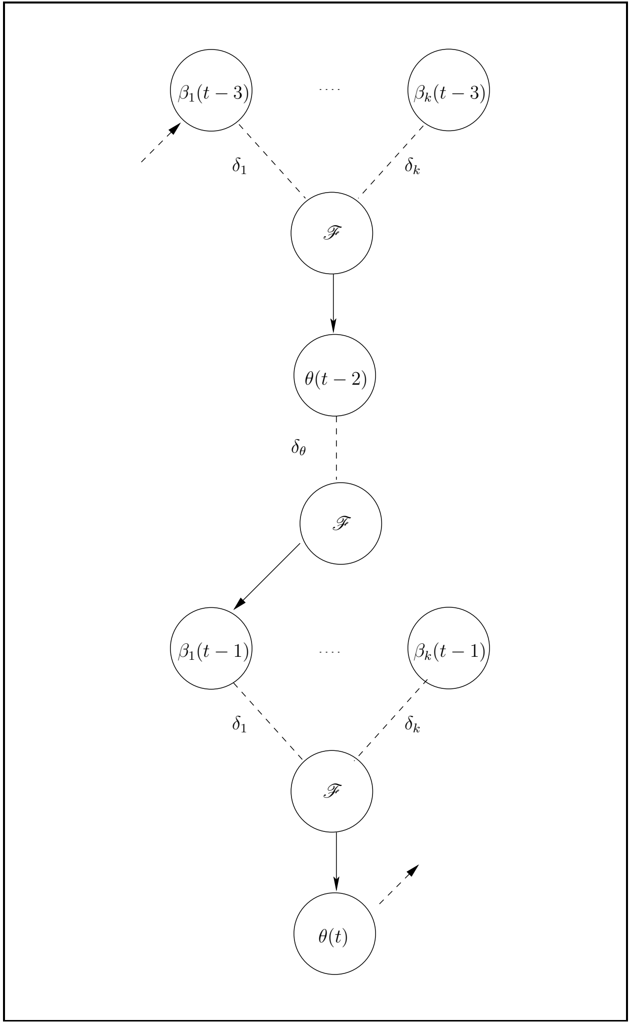


Figure 2: A temporal F-model "unrolled" in time: each $\delta_i = 1$ time unit, and the cycle period $= 2$ time units.

In high dimensional spaces such as microarray data many independent processes may run concurrently and there may exist various causal chains, each with different cycle periods. Hence assuming a global cycle period on all variables of the system is too restrictive. Therefore we use a *relative* cycle period for each variable.

Definition 3 ***(Cycle period)*** *Let* $X(t) = \langle x_1,\dots,x_m\rangle$ *be an expression vector and*

$\lambda \in \{1, \ldots, m\}$. *Then the cycle period for $X(t)$ is the first off-zero peak correlation, i.e., the first λ for which*

$$\arg\max_{\lambda \in \{1,\ldots,m\}} \left(\sum_{t=1}^{m-\lambda} X(t) \cdot X(t+\lambda) \right) > 0$$

As mentioned in Section 2 we use discretized data. In (Friedman, Linial, Nachman & Pe'er 2000) a discretization was used where any log expression ratio outside a central band of ±0.5 was taken as gene activity. However, this will tend to ignore low-amplitude gene expressions which could form part of regulatory interactions, e.g., certain transcription factors. Therefore we discretize each expression vector separately using a relative threshold: expression values greater (resp. less) than the threshold are mapped to 1 (resp. -1), otherwise they are mapped to zero. The relative threshold is set via a k^{th} order statistic, where k is a parameter to the algorithm.

Finally, the compression function $\mathscr{F}$ combines a set of variables (candidate causes) into a single variable. In application to microarray data the variables are time-shifted, discretized gene expression profiles representing candidate regulators of a selected target gene. For our implementation we used the following compression function:

Definition 4 *Given a set of expression values $X \equiv \langle x_1, \ldots, x_n \rangle$ for n genes at some point in time the compressed version of X is:*

$$\mathscr{F}(x_1, \ldots, x_n) = \begin{cases} +1, & \textit{if } |X^+| \geq n - g \\ -1, & \textit{if } |X^-| \leq n - g \\ 0, & \textit{otherwise} \end{cases}$$

where $X^+ = \{x \in X \mid x = +1\}$, and $X^- = \{x \in X \mid x = -1\}$ and $0 \leq g < n/2$ is a slack parameter to allow some variation in $+$ or $-$.

The slack parameter is to allow for noise. This compression function is applied to all columns in the set of candidate causes, giving a compressed variable. Note that as the number of variables n increases, the compressed values either remain at $+1$ or -1, or go to zero. This helps to avoid overfitting of the model: adding too many variables as causes will tend to reduce the compressed value to zero, which implies the model loses any dependency between the candidate causes and the effect. We only add genes as candidate causes whose expression profiles are closely similar to the target gene. Further details on the compression function $\mathscr{F}$ are in Ahsan (2006).

3 A local learning algorithm

The key problem investigated here may be formalized as: *Given a data set D, find an $\mathscr{F}$-model which adequately explains D.* Since computational efficiency is a key issue for microarray data which has many variables we propose searching for *local neighborhoods* instead of searching for an entire network, as conditional independence may be efficiently computed.

For directed graphs this problem simplifies to searching for the immediate parents of a variable. Various definitions of an immediate parent have been used, e.g., (Pearl 1988, Margaritis & Thrun 2004, Koller & Sahami 1996). For us, however, immediate parents (a) temporally precede the target, (b) significantly correlate with the target, and (c) render the target independent of all its ancestors. Based on this definition we propose the following local learning algorithm.

Algorithm 1 learns causes of an effect, and is divided into two phases. Phase 1 filters out a set of plausible candidates based on a pairwise score between each candidate and the effect (lines 3-5). The score used was the mutual information $I(X;Y)$ between a pair of variables (Kullback & Leibler 1951).

Algorithm 1 Learning Immediate Causes of an Effect: PIA

```
Input: Initial candidates $\Theta$, Effect $\theta$ and Threshold
       $\tau$ for score
Output: Set of immediate causes for $\theta$
 1: PHASE 1: Selecting a Set of Candidate Causes
    for $\theta$
 2: $\alpha \leftarrow \{\}$
 3: for $\beta \in \Theta$ do
 4:   if $I(\theta;\beta) > \tau$ then
 5:     $\alpha \leftarrow \alpha \cup \{\beta\}$
 6: PHASE 2: Learn local neighborhood of $\theta$
 7: Sort $\alpha$ according to $\succ_\theta$      //see text for details
 8: $\alpha' \leftarrow \{\}$
 9: //Seek causes that increase the score given the
    current $\alpha$
10: repeat
11:   $\beta \leftarrow max(\alpha)$
12:   if $I(\theta;\mathscr{F}(\alpha',\beta(t-\delta_{\theta\beta}))) - I(\theta;\mathscr{F}(\alpha')) > 0$
      then
13:     $\alpha' \leftarrow \alpha' \cup \{\beta\}$
14:   $\alpha \leftarrow \alpha - \{\beta\}$
15: until $\alpha = \emptyset$
16: return $\alpha'$
```

In phase 2 the algorithm consists of two further stages. At line 7 the algorithm first sorts the filtered candidates α with respect to the following order: $\beta_1 \succ_\theta \beta_2$ iff either $\delta_{\theta\beta_1} < \delta_{\theta\beta_2}$, or $\delta_{\theta\beta_1} = \delta_{\theta\beta_2}$ and $I(\theta,\beta_1) > I(\theta,\beta_2)$. This ordering relies on computation of *delays* or time shifts δ_{XY} between variables X and Y. The time shift between a pair of variables is simply the time difference between the respective peaks in each time series. Sorting in this way implements criteria (a) and (b) of the definition of immediate parents.

The algorithm then iteratively processes each cause according to the order $\succ_\theta$ in a greedy manner (lines 10-15). At each step the current candidate cause β is added to the set of immediate causes of θ if this results in an increase in the mutual information score between θ and the updated compressed variable $\mathscr{F}(\alpha',\beta(t-\delta_{\theta\beta}))$. This test implements criterion (c) of the definition of immediate parents. The notation $\beta(t-\delta_{\theta\beta})$ denotes the application of a variable-specific time shift before computing the score. Note that our selection procedure is biased towards variables with shorter time delays; the rationale for this is that longer time delays can be caused by a chain of causes in the network, whereas shorter ones cannot.

The temporal $\mathscr{F}$-model is based on the assumption that the time shift for any cycle in an $\mathscr{F}$-model is a multiple of the period λ, and that any path with the time shift greater than λ must include a cycle. However, target variables may vary in λ, and therefore simply piecing all local neighborhoods will not result in a true $\mathscr{F}$-model. Currently, no constraint has been enforced on the local learning algorithm PIA which would allow one to induce a global structure in a modular manner. However, one may induce $\mathscr{F}$-models by grouping target variables with similar cycle periods, and post-pruning any edges that conflict with cyclicity assumptions made in Section 2.2. This approach, called the Piece-wise Network Induction Algorithm (PWNIA), was used for the experiments in the next section.

4 Experimental results

A major issue in current research on learning genetic regulatory networks from genome-wide data is that there are no reference "ground truth" models. However, we attempted to validate our approach in two ways: by reconstruction experiments using simulated data; and on real 0 data by using selected sets of genes and evaluating features of learned network structures against known properties of the cellular system.

4.1 Simulated data

We conducted simulation experiments by implementing a simple model of cell-cycle gene expression. Input to the simulator was in the form of a directed graph, modelling a genetic regulatory network, plus properties of each edge in the graph, representing a gene interaction, such as cycle length, time delays, etc. For each edge $A \rightarrow B$ in the graph, the dependence of B on A was modelled as a probabilistic, time-delayed function of A, with added Gaussian noise. Variables with no parents (root nodes) were modelled as sine functions with added Gaussian noise. The simulator was implemented to enable manipulation of a number of parameters, such as the amount of added noise, the sampling frequency of data, etc.

Graphs were generated randomly using the preferential attachment model of Albert and Barabasi (1999) implemented as part of the Python random graph library NetworkX [1]. The edges of these undirected graphs were directed arbitrarily, avoiding cycles.

For each randomly-generated graph and a set of associated probabilistic dependencies, the simulator was used to generate multiple time series data sets to which the PWNIA algorithm was applied to learn a network. Learned networks were then compared with the originals to evaluate the reconstruction. Features of the reconstructed networks were evaluated using a version of the approach in Friedman et al. (1999) adapted for time series data.

We investigated the effect on learning of (a) adding varying amounts of noise, (b) varying the number of time points, (c) varying network density, and (d) varying parameters of the learning algorithm. Precision and recall measures were recorded for recovery of order and Markov relations (Friedman, Goldszmidt & Wyner 1999). Order relations are true for gene-gene interactions $A \rightarrow B$ where A is an ancestor of B. However, Markov relations are more stringent, containing only the subset of genes that make a gene probabilistically independent of all other genes in the network. Note that precision and recall curves were generated as opposed to the use of either sensitivity/specificity or ROC analysis due to the difficulty of generating true negatives for order and, particularly, Markov relations in our experimental setup.

In terms of precision; which reflects the extent to which predicted gene-to-gene relationships made by our learning algorithm were correct, our results were similar for both order and Markov relations over experimental conditions (a) to (d). However, for recall; the extent to which actual relationships were predicted by our algorithm, we found that order relations were much easier to reconstruct than Markov relations.

For example, condition (a), adding noise to the generation of simulated data, led to reductions in precision (from 0.9 to 0.6 for order relations and from 0.85 to 0.65 for Markov relations) and greater reductions in recall (from 1.0 to 0.7 for order relations and from 0.45 to 0.25 for Markov relations), all measures are rounded mean values for 30 data sets. Since recall for order relations was higher than for Markov relations, it follows that it should be easier on real data to discover general causal ordering relations than more detailed gene-to-gene regulatory interactions. This formed the basis for the experiments on real microarray data discussed below.

Real microarray data contains typically only a few time points, since each time point represents a separate microarray experiment, which is time-consuming and relatively expensive to carry out. With simulated data, however, the number of data points measured can be varied. We found that under experimental conditions (b) and (c), increasing the number of data points per cycle, led to increases in both recall and precision, except for precision on reconstruction of the denser networks. This was as expected, since a chance correlation is more likely because more variables have shorter time shifts and hence appear near the top of the order $\succ_\theta$, leading to possible inclusion as immediate causes. Experimental condition (d), varying the similarity threshold above which genes are considered potential parents, showed an inverse relationship for both order and Markov relations, i.e., recall fell while precision rose as the threshold was increased. This suggests that the threshold may be acting to control the number of false positives, although at the expense of coverage. If so this could be a useful property of the algorithm and should be investigated further.

An additional parameter, not investigated due to lack of time, that may be useful in reducing the effect of noise on the algorithm is the *slack parameter*. This has the effect of relaxing the strict requirement of direct or inverted similarity between a gene and its candidate regulators. As part of future work this should be investigated in combination with the similarity threshold, as the former may compensate for the reduction in recall due to the application of the latter. Lastly, we have implemented a fixed form of compression function $\mathscr{F}$. This may not be the most appropriate for for learning genetic regulatory interactions, and it would be interesting to investigate the possibility of learning the compression function from data.

The simulation experiments are described in more detail in Ahsan (2006). We concluded from these results on simulated data that the algorithm shows promise for the discovery of gene relationships, although it is likely to be susceptible to noise and low numbers of instances.

4.2 Yeast cell cycle data

Since it is not currently possible to evaluate network learning attempts against a "real" genetic regulatory network, even in well-studied organisms such as the budding yeast *Saccharomyces cerevisiae*, on real data we took the approach of examining key *temporal* features. At this stage we are only attempting to validate our approach rather than generate biologically useful knowledge. Therefore, following the results discussed above on learning order relations on simulated data we investigated the extent to which the edges in a learned network reflect known temporal features of the domain.

Our starting point was the seminal experimental work on the budding yeast cell cycle by Spellman et al. (1998). In this work the cell cycle was arrested by different means and on release the cells went through one to two synchronised cell cycle iterations, during which microarray gene expression measurements were take at regular intervals. Using a Fourier analysis-based scoring function, calibrated to known cell-cycle regulated genes, 800 genes were determined as being cell-cycle regulated in terms of their gene expression. In addition, a temporal ordering was applied to sort these 800 genes into one of five cell cycle *phases*, G1, S, S/G2, G2/M or M/G1. This provides a benchmark, similar to the order relations from our simulated data, against which to evaluate our algorithm.

The objective is as follows: given a yeast cell-

[1] https://networkx.lanl.gov

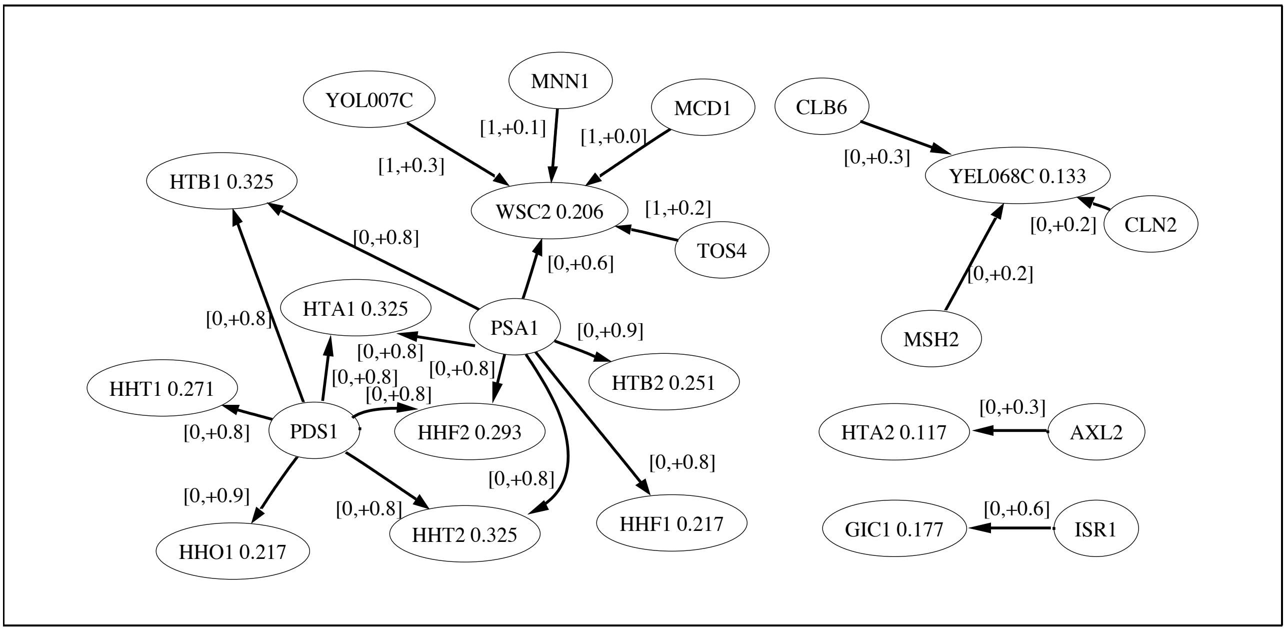


Figure 3: A network learned from the top-scoring 135 genes on the alpha cell cycle data set. Each node with an in-arrow is labelled by its mutual information with the compressed parent set. Each edge from parent to target represents a G1 to S phase interaction and is labelled with 2 numbers: time shift and pairwise correlation coefficient (see text for details).

cycle microarray time series data set, run the network learning algorithm and compare the edges thus obtained to the known temporal ordering in terms of their *phase coherence*. Phase coherence is defined by a set of rules stating the biological plausibility of each possible phase labelling of edges in a network graph. For example, if the cause and effect were both in the same phase, this would be acceptable, whereas if the cause and effect were separated by a large number of phases, this is unlikely to be a biologically valid causal relationship.

The rules were:

Cause phase	*Possible effect phases*
G1	S, S/G2 or G2/M
S	S/G2, G2/M or M/G1
S/G2	G2/M or M/G1
G2/M	M/G1 (current cycle) or G1 (next)
M/G1	G1 (next cycle) or S (next cycle)

This defines 12 out of a possible 25 cause-effect phase relations. Allowing the 5 same-phase relations gives a total of 17 permitted by these rules. Notice that the last two phases in the cell cycle, G2/M and M/G1, have causal relations that cross the cell-division boundary.

To test the phase coherence of the 0 learned by our algorithms we performed a number of network learning experiments using three yeast cell-cycle time series data sets from the work of Spellman et al. (1998). Since our algorithm requires microarray data containing reasonable cyclic expression profiles, we restricted attention to the 800 genes determined to be cell cycle regulated. We ranked these in decreasing order of their aggregate score – known as the "CDC score" – generated by Spellman et al. (1998). The higher the CDC score, the clearer the periodic "signal" in the microarray data. This score is calibrated to genes known to be cell-cycle regulated, and ranges from 15.990 (maximum) to 1.314 (minimum). We set two thresholds on this score: ≥ 5.0 and ≥ 3.0. There are 135 genes above the first threshold and 297 genes above the second. This gave us the basis to construct three sets of genes, in decreasing order of "cyclicity", of size 135, 297 and 800.

We ran our network learning algorithm with default parameter settings for the three gene sets on each of the three microarray data sets known as *alpha*, *cdc15* and *cdc28*. (We did not use the *elutriation* data set since it does not comprise two complete cell cycles and our algorithm currently requires ≥ 2 cycles.) The set of edges from each learned network was filtered to select only those having a Pearson pairwise correlation above a certain correlation threshold (results shown are for $r \geq 0.7$). Nodes in each set of "well-correlating" edges was then labelled their respective phase. This phase labelling was then evaluated for coherence against the above rules.

Data		135	297	800
alpha	corr.	0.75 (239)	0.73 (508)	0.63 (1154)
	legal	0.91 (180)	0.90 (369)	0.85 (726)
cdc15	corr.	0.85 (217)	0.78 (498)	0.56 (1592)
	legal	0.91 (184)	0.89 (387)	0.83 (888)
cdc28	corr.	0.71 (224)	0.70 (490)	0.59 (1383)
	legal	0.91 (154)	0.89 (344)	0.87 (816)

Table 1: Results from phase coherence tests on edges from learned networks on 3 cell-cycle regulated data sets. Shown are the proportions (totals) of well-correlating edges and phase-coherent edges for 3 gene sets of increasing size and reducing overall quality (see text for details).

The results are summarised in Table 1. For each data set there are two rows, labelled "corr." and "legal". The "corr." row contains the proportion (total in brackets) of edges above the correlation threshold. The "legal" row contains the proportion (total in brackets) of edges that are phase coherent, i.e., are in accord with the rules above. The three columns refer to the three different-sized gene sets.

The results show that the algorithm is constructing network graphs containing a majority of well-correlated edges. In addition, the phase-coherence of the edge sets remains high even on the largest data set (800) which produces networks with many low-correlating edges. Note that the pairwise correlation is not the necessarily a good measure of functional relatedness, since it ignores multi-gene dependencies, unlike the mutual information score used by our algorithm. However, we use it here as a useful heuristic, since it could easily form a pre-processing step for our algorithm. Note also that the algorithm is *not* given the phase information – phase coherence is used solely to assess performance.

We are continuing to work on examining the phase coherence of this approach. For example, we can isolate genes from different phases to use as candidate cause and effect sets to initialise our algorithm. The edge sets can then illustrate how well temporal ordering is being recovered during learning. Figure 3 shows one network, learned by setting the candidate causes to G1-phase genes in the top-scoring 135 and the candidate effects to S-phase genes from the same set. So far we have not compared the performance of this algorithm with alternative approaches, but this should be undertaken as part of future work. We believe that ideas such as phase coherence may provide useful methods to compare performance of such algorithms in the absence of a "gold standard".

5 Conclusions

In this paper we have developed a novel framework for modelling and learning causal gene-to-gene interactions from microarray data. Our approach extends the standard Bayesian network formalism to allow temporal relations and enable cyclic dependencies which is a critical requirement for the representation of biological regulation. Some key simplifying features of the approach are that gene expression is discretized, and each gene has a set of immediate causes, with an associated time shift, on its gene expression. Furthermore, the effect of multiple causes is combined into a single variable, also discretized, via a novel compression function.

The dependence of the gene on its compressed parent, which precedes it in the temporal ordering, is modelled with a simple discrete probability distribution. Our learning algorithm focuses on identifying the immediate causes of a given gene. The compression function limits the introduction of new parents through its behavior in losing information with more parents, and fixes the number of parameters to be learnt (avoiding over-fitting).

Experimental validation is promising, based on simulated and real data. On real data we demonstrated that the networks learned are largely consistent with the temporal ordering properties of the cell cycle, reinforcing some results from experiments on simulated data.

The extent to which learned interactions are spurious (false positives) is not known. This is a key area for future work. We believe that combining microarray data with data from other sources, such as protein-protein interactions, has potential for improving this aspect of our approach. For example, some critical aspects of function are clearly absent from microarray data. An example is the gene CDC28, which is central to cell-cycle regulation, but assumed to be expressed in excess and therefore will not be learned as part of any interaction from microarray data alone, although it is well-represented in protein interaction databases.

References

Ahsan, N. (2006), Learning Causal Networks from Gene Expression Data (Submitted), Master's thesis, University of New South Wales.

Barabasi, A. L. & Albert, R. (1999), 'Emergence of scaling in random networks', *Science* **286**(5439), 509 – 512.

de Jong, H. (2002), 'Modeling and Simulation of Genetic Regulatory Systems: A Literature Review', *Journal of Computational Biology* **1**(9), 67–103.

Endy, D. & Brent, R. (2001), 'Modelling cellular behaviour', *Nature* **409**, 391–395.

Friedman, N., Goldszmidt, M. & Wyner, A. (1999), Data analysis with bayesian networks: A bootstrap approach, *in* 'Proceedings of UAI99', pp. 196–205.

Friedman, N., Linial, M., Nachman, I. & Pe'er, D. (2000), Using Bayesian networks to analyze expression data, *in* 'RECOMB', pp. 127–135.

Hume, D. (1999), *Enquiry concerning Human Understanding*, Oxford/New York: Oxford University Press.

Koller, D. & Sahami, M. (1996), Toward optimal feature selection, *in* 'International Conference on Machine Learning', pp. 284–292.

Kullback, S. & Leibler, R. A. (1951), 'On information and sufficiency', *Ann. Math. Statist.* pp. 22:79–86.

Margaritis, D. & Thrun, S. (2004), 'Bayesian network induction via local neighborhoods', *Advances in Neural Information Processing Systems 12* .

Pearl, J. (1988), *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*, Morgan Kaufmann.

Spellman, P., Sherlock, G., Zhang, M., Iyer, V., Anders, K., Eisen, M., Brown, P., Botstein, D. & Futcher, B. (1998), 'Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast Saccharomyces cerevisiae by Microarray Hybridization', *Molecular Biology of the Cell* **9**, 3273–3297.

Bacterial promoter modeling and prediction for *E. coli* and *B. subtilis* with Beagle

Stefan R. Maetschke **Michael W. Towsey** **James M. Hogan**

Faculty of Information Technology, Queensland University of Technology, GPO Box 2434, Brisbane, QLD 4001, Australia
Email: m.towsey@qut.edu.au

Abstract

We constructed σ^{70}-promoter models of varying complexity to predict promoter locations and to evaluate the importance of specific promoter elements. For this purpose, a novel software, named *Beagle*, was developed that utilizes an easy description language to conveniently specify promoter models. Model specifications are translated into position weight matrices and gap distributions which are refined using data from known promoters.

The method is transparent, fast and allows the rapid exploration of different promoter models. Applied to promoter prediction in *E. coli* and *B. subtilis*, we show that inclusion of UP-elements and extended -10 motifs into the model yields a significant increase in prediction accuracy.

The software, data sets and extended results can be downloaded at http://eresearch.fit.qut.edu.au/Beagle/.

Keywords: Beagle, Promoter, sigma-70, Escherichia coli, Bacillus subtilis

1 Introduction

Promoters are regions of DNA responsible for the initiation of gene transcription. Their identification is crucial for understanding gene regulation but they are difficult to identify *in silico* because their important functional sites are poorly conserved. Identifying promoters using wet-lab techniques is time consuming and given the exponentially growing number of sequenced genomes, there is a definite need for computational methods to detect and study promoters.

Many methods have been devised to identify promoter sites using for example, Regular Expressions (REs), Position Weight Matrices (PWMs), Hidden Markov Models (HMMs), Neural Networks (NNs) and Support Vector Machines (SVMs) (Vanet, Marsan & Sagot 1999). The different model types have strengths and weaknesses which typically involve trade-offs between accuracy, transparency, speed and ease of use. Despite (or perhaps because of) their simplicity, PWMs continue to be a frequently used approach to search for promoters. In addition their use finds theoretical justification in Information Theory (Schneider, Stormo, Gold & Ehrenfeucht 1986).

PWMs have been used in two ways to search for promoters. The direct approach is to search for DNA motifs that bind the RNA Polymerase (RNAP) holoenzyme. In the case of the σ^{70} family of bacterial promoters, with which we are solely concerned in this paper, this means having PWM definitions for two binding sites located at -35 and -10 base pairs (bp) with respect to the Transcription Start Site (TSS). The difficulty with this direct approach is that the known binding sites are highly variable, leading to a high rate of false positive predictions for a satisfactory rate of recall.

The indirect approach to promoter prediction depends on the observation that promoters are accompanied by other binding sites for transcription factors which modulate transcription. Given access to a sufficiently large number of definitions of known transcription factor binding sites (TFBSs), clusters of high scoring hits indicate the presence of a promoter. For example, the well known MatInspector (Cartharius, Frech, Grote, Klocke, Haltmeier, Klingenhoff, Frisch, Bayerlein & Werner 2005) and Cluster-Buster (Frith, Li & Weng 2003) programs both use this strategy which is particularly useful with eukaryotic organisms.

As more becomes known about the structure and function of bacterial RNAP, it is clear that the enzyme interacts with the DNA double helix in more complex ways than just the canonical -10 and -35 interactions (Mitchell, Zheng, Busby & Minchin 2003, Miroslavova & Busby 2006). The purpose of this paper is to revisit the direct approach to identifying bacterial promoters but to build models that incorporate more of what we have recently learned about the DNA-RNAP interaction. To this end, we have developed a software tool, *Beagle*, that utilizes a simple description language to specify bacterial promoter models. Internally, the models are realized as a sequence of PWMs and gap length distributions. The model parameters are refined using experimentally confirmed TSSs. *Beagle* achieves good accuracy compared to more complex machine learning methods but is faster to train and easier to use. In addition, the generated models are transparent and permit direct biological interpretation.

This paper is organized as follows: In Section 2 we discuss related supervised learning algorithms for promoter prediction. The biological background that drives our promoter models is provided in Section 3 and the data utilized to evaluate various models are described in Section 4. Section 5 explains some of the algorithmic detail behind *Beagle*. Prediction results are presented in Section 6 followed by the conclusion in Section 7.

2 Related work

Many methods have been developed for promoter prediction. Vanet *et al.* (Vanet et al. 1999) provides a good overview of the various approaches. We focus our attention on three more recent contributions to the literature that offer interesting comparisons with our work.

Huerta *et al.* (2003) derived PWMs for the -35 and -10 elements of σ^{70} promoters in *E. coli* from multiple alignments of known promoters. The PWMs were optimized using information content and similarity to a known consensus. Typically their derived PWMs extended two or more bases upstream of the canonical -10 and -35 hexamers and their models also incorporated scores derived from frequency of spacer lengths and distance to the gene start site (GSS). They observed that true promoters tend to occur in regions where there is a cluster of high scoring putative promoters. And in about 50% of cases, the true promoter was not the highest scoring location.

Gordon *et al.* (2006) trained an ensemble of Support Vector Machines (SVMs) for bacterial promoter prediction using a variant of the mismatch string kernel. The SVM approach was more accurate than the PWM approach but highest accuracy was obtained with a model that combined scores from the ensemble-SVM, PWMs and GSS to TSS distance. An obvious drawback with an ensemble of 40 SVMs is the time required to train them – typically several orders of magnitude more than the estimation of parameters for PWM models.

Burden *et al.* (2005) trained a series of Time Delay Neural Networks (TDNNs) to model multiple promoter elements. They demonstrate greatly improved accuracy when distance to GSS is incorporated into the models. However the number and type of model elements was fixed and TDNNs are typically time consuming to train.

The primary motivation for Beagle is the explicit incorporation of additional DNA motifs into promoter models based on our emerging understanding of the action of RNAP. Beagle gives the experimenter control over all elements of the promoter model, enabling a variety of hypotheses to be tested. While the PWM models of Huerta *et al.* (2003) included extended -10 and extended -35 elements, they were not user defined and it was not demonstrated how these contributed to prediction accuracy. In the case of the ensemble-SVM approach, Gordon *et al.* (2006) identified DNA locations important for classification accuracy. Not surprisingly the -10 and -35 locations were most important but also the ribosomal binding site motif figured strongly around the +20 location, indicative of the fact that most promoters lie close to their GSS. Locations upstream of the -35 box and an extended -10 were not identified as important for classification but the method had limited resolution.

3 Biological Background

Bacterial RNAP is a protein complex composed of five subunits, $\alpha_2\beta\beta'\omega$ (Murakami & Darst 2003). To initiate transcription, the core enzyme must first acquire an additional σ subunit whose function is to recognize a promoter (Gross, Chan, Dombroski, Gruber, Sharp, Tupy & Young 1998). DNA binding initiates a series of structural changes that result in DNA strand separation at the -10 site. After several cycles of formation and release of short transcripts, the σ-factor dissociates and gene transcription commences (Murakami & Darst 2003).

It has long been known that domains 2 and 4 of the σ factor bind to the strongly conserved -10 and -35 boxes. More recently, it has been demonstrated that a third domain interacts with a so-called *extended* -10 element (see Fig. 1) (Miroslavova & Busby 2006). First identified in *B. subtilis*, the extended -10 element is also present in about 20% of *E. coli* promoters. It is located three base pairs upstream of the -10 element with consensus `TG` (Mitchell et al. 2003). Mitchell *et al.* (2003) also identified the importance of a longer extended -16 region (consensus `TRTG`[1]), which is important for some *E. coli* promoters. *In vitro* experiments have demonstrated that domain 3 interaction with an extended -10 or -16 consensus site can compensate for weaker -10 or -35 interactions but that a combination of consensus -10, extended -10 and -35 motifs reduces gene expression (Miroslavova & Busby 2006).

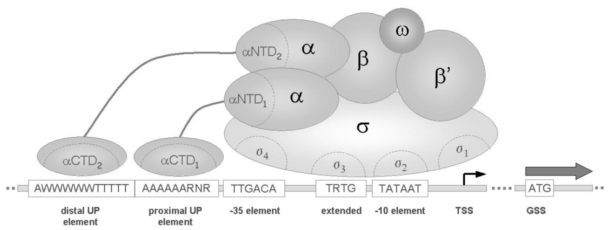


Figure 1: Schematic diagram of the RNA polymerase holoenzyme and its binding elements within the promoter region.

The α subunits also play a key role in the initiation of transcription. Each consists of two domains connected by a flexible linker. The amino-terminal domains (αNTD) form part of the main body of the holoenzyme, while the carboxy-terminal domains (αCTD) are free to interact with UP-elements and activators (Estrem, Ross, Gaal, Chen, Niu, Ebright & Gourse 1999).

An UP-element is an `A/T` rich region about 20 bp long located immediately upstream of the -35 element. Each of the two αCTD domains can bind autonomously to the proximal or distal part of an UP-element (Typas & Hengge 2005). It has been shown for some promoters that interactions between one or both α subunits and the UP-elements can increase promoter activity by a factor of 10 or more (Estrem et al. 1999).

The focus of this paper is to determine whether incorporation of these more recently discovered functional sites into promoter models improves the prediction of σ^{70} dependent promoters.

4 Data set

For our experiments we utilized the bacterial genomes of *Escherichia Coli* K-12 MG1655 (ACCN:U00096.2)[2] and *Bacillus subtilis* (ACCN: NC_000964.2)[3].

Experimentally confirmed TSS locations for *E. coli* were obtained from the RegulonDB database[4]. The data set was filtered for unique σ^{70}-promoters with known TSS locations, resulting in 542 records. We then determined the genes in *E. coli* closest to the given TSS locations and extracted the corresponding upstream regions. Following Huerta *et al* (2003), we eliminated all upstream regions (USRs) with a TSS location further than 250 bp from the gene start. The final data set for *E. coli* consisted of 492 sequences, each containing a single annotated TSS location.

A list of TSS locations for *B. subtilis* was was obtained from DBTBS (Release 4)[5], a database of transcriptional regulation in Bacillus subtilis. This list contains 275 TSS predictions from which we selected 205 that were within 250 bp upstream of the nearest gene start site.

[1] `N` = any nucleotide, `R` = `A` or `G` and `W` = `A` or `T`, according to the IUPAC DNA alphabet.

[2] `ftp://ftp.ncbi.nih.gov/genbank/genomes/Bacteria/Escherichia_coli_K12/U00096.gbk`

[3] `http://www.ncbi.nlm.nih.gov/entrez/viewer.fcgi??db=nucleotide&val=NC_000964`

[4] `http://regulondb.ccg.unam.mx/data/PromoterSet.txt`

[5] `http://dbtbs.hgc.jp/COG/tfac/SigA.html`

5 Beagle

Beagle builds promoter models in two steps. The first step involves initialization of the model using a simple promoter description language and the second step refines the model iteratively. The final model consists of a series of optimized PWMs and gap length distributions.

The initialization phase takes as input a promoter description string which defines a set of consensus motifs and the gaps between them. For instance, the canonical model of a σ^{70}-promoter has a -35 `TTGACA` element, a 15-21 bp spacer, a -10 `TATAAT` element and a 4-13 bp discriminator culminating in the TSS. This canonical promoter can be specified in Beagle by the description string:

```
<TTGACA (15,21) TATAAT (4,13) TSS>
```

A promoter description can contain an arbitrary number of binding motifs and gap definitions. In particular, models can include the gap between TSS and GSS and incorporate UP elements and extended -10 motifs.

Beagle parses the description string and translates it into a model composed of PWMs and weighted gaps. In the initialization step, the PWM elements are set to represent the required consensus sequences and the gap length frequencies are initialized to a uniform distribution.

The model parameters are optimized during a training phase using an iterative bootstrap approach. At each iteration, the model's TSS position is anchored to the known TSS position of a training sequence and, by exhaustively scoring all valid arrangements of PWM matches taking the current gap distribution into account, the highest scoring combined match is found. Gap weights also contribute to the score[6]. To generate an improved model, maximum likelihood estimates for new PWM and gap weights are calculated from the best match in each of the training sequences. This bootstrapping process continues iteratively until the information content of the PWMs ceases to increase.

For prediction, the model TSS is anchored at each position of the query sequence and the score of the best match is given to that position. The position with the highest overall match score becomes the predicted, putative TSS for that sequence. For more details see the manual which accompanies the software download.

The initial promoter description string may also incorporate a marker for the gene start site (GSS). This permits the definition of models that take the distance to the downstream GSS into account. The GSS marker is always anchored to the nearest gene start site and the weights for the distribution of TSS-GSS gaps are evaluated in exactly the same way as for other gap/spacers in the model. Gaps have a so called *impact factor*, which weights the relative contribution of the gap score to the overall model score. In the following model of a canonical promoter with extended -10 and TSS-GSS gap, gap scores contribute 20% to the overall score:

```
TTGACA (12,18,0.2) TGNTATAAT (4,13,0.2) TSS (0,249,0.2) GSS
```

The overall match score s_{all} of a sequence to a model consisting of N elements (PWMs or gaps) with element scores s_i and impact factors f_i, is calculated as follows:

$$s_{all} = \frac{\sum_i^N f_i \cdot s_i}{\sum_i^N f_i}, \quad \text{with } s_i, f_i \in \{0,1\}. \qquad (1)$$

Beagle has some similarity to Meta-MEME (Grundy, Bailey, Elkan & Baker 1997) in that the required patterns are modeled as a set of conserved motifs separated by gaps. But where Meta-MEME uses MEME to obtain an initial PWM description of the conserved motifs, Beagle derives its PWM description from a user supplied consensus. And whereas Meta-MEME then embeds the PWMs into a Hidden Markov Model along with a probabilistic description of the gaps, Beagle preserves the PWMs and gaps as discrete entities.

In the next section, we demonstrate the performance of various promoter models for TSS prediction.

6 Results

We used Beagle to explore extensions to the canonical promoter model by incorporating various combinations of (1) the extended -10 element (consensus `TG`), (2) the -16 element (consensus `TRTG`), (3) UP-elements and (4) distance between TSS and GSS (see Fig. 1). We experimented with three different UP-element sequences that appear to be prominent in several *E. coli* and *B. subtilis* promoters: (1) The most general UP-element is an `A/T`-rich region described in our description language as `NNWWWWWWWWWWWWWWWWWNN`. (2) For the promoter *rrnB*-P1 in *E. coli*, Estrem *et al.* (Estrem, Gaal, Ross & Gourse 1998) reported an UP-element with the consensus sequence `NNAAAWWTWTTNNAAANNN`. (3) According to Gourse *et al.* (2000), UP-elements can be divided into a more important proximal motif (`AAAAAARNR`) and a distal motif (`NNAAAWWTWTTN`). We incorporated the proximal half of the motif only.

Table 1 shows the prediction accuracies for a variety of promoter models when applied to two sets of known promoters in *E. coli* and *B. subtilis*. The result for the canonical promoter (`TTGACA (15,21,0.2) TATAAT (4,13,0.2)` is shown in the top left of each table. Prediction accuracy is calculated as the percentage of predicted TSS locations that are at most ±5 bp from the true TSS[7]. Interpretation of results can be helped by reference to Fig. 2 which illustrates the sequence logos obtained from training data for the most successful model in each genome.

It is immediately apparent that prediction accuracies are up to 50% higher for *B. subtilis* promoters than for *E. coli* promoters. The sequence logos in Fig. 2 illustrate that the *B. subtilis* promoters have higher information content and are more highly conserved. It must also be the case that a larger fraction of *B. subtilis* TSSs are located at the highest scoring location upstream of their genes than is case for *E. coli* promoters. *B. subtilis* has 18 identified sigma factors compared with seven known for *E. coli*. It is thought that this is due to the greater regulatory demands placed on *B. subtilis* given its more variable soil environment. We might expect that having more σ-factors requires *B. subtilis* to conserve the differences between them by keeping binding sites closer to the consensus.

Another interesting difference between the two species is that inclusion of the TSS-GSS distance in the initial promoter definition improves prediction accuracy significantly in *E. coli* but not in *B. subtilis*. Again this can be explained if a larger fraction of *B. subtilis* TSSs are located at the highest scoring location upstream of their genes no matter how far upstream.

The effect of including an UP-element in the promoter definition (in the absence of an extended -10

[6] Beagle utilizes the BioPatML pattern matching engine for this purpose. See `http://eresearch.fit.qut.edu.au/BioPatML/` for details.

[7] There is no consistent definition of a true positive TSS prediction in the literature. We follow the definition of Huerta *et al* (Huerta & Collado-Vides 2003).

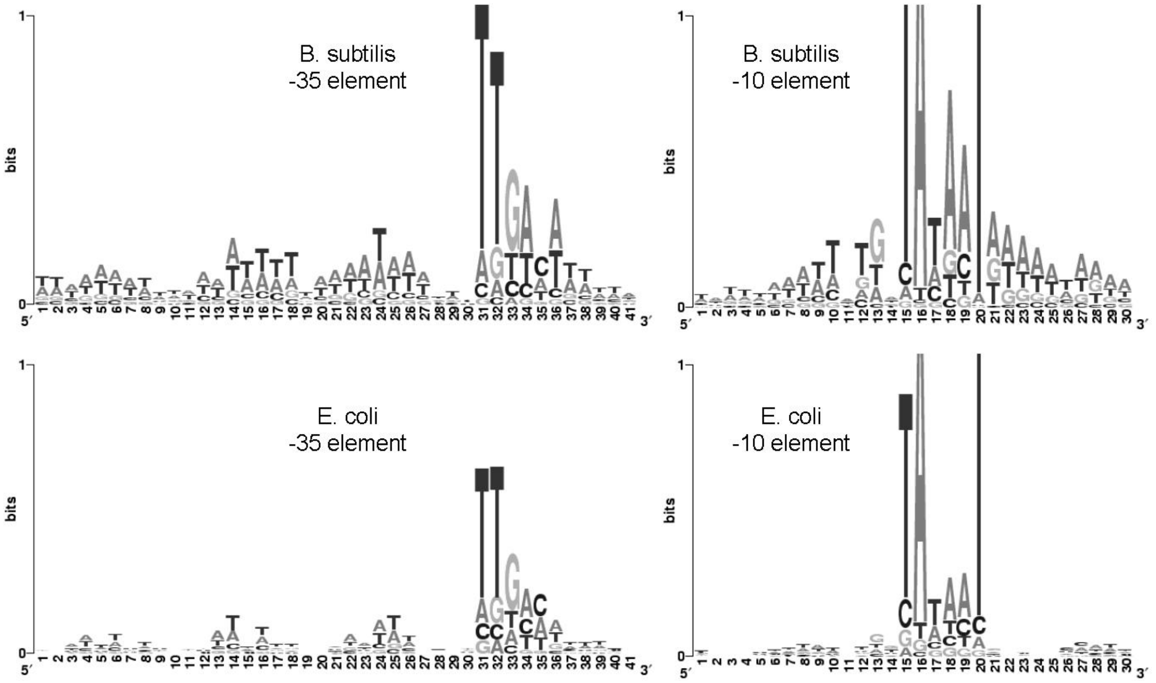


Figure 2: Logos of the vicinity of the -35 and -10 elements of the best performing promoter model in *E. coli* and *B. subtilis*. Note that the y-axis scale has been truncated to 1 bit in order to highlight detail in the upstream region. Logos created with WebLogo at `weblogo.berkeley.edu`.

motif) was variable. The fully defined UP-element NNAAAWWTWTTNNAAANNN had a deleterious effect on prediction performance while the `A/T`-rich UP-element `NNWWWWWWWWWWWWWWWWWWNN` and the proximal UP-element (`AAAAAARNR`) both improved prediction accuracy.

In *E. coli*, use of the extended -10 (`TRTG`) had a deleterious effect on promoter prediction in all cases. Interestingly, use of the `TG` extended -10 also had a deleterious effect on prediction accuracy except when used in conjunction with the `A/T`-rich UP-element. This interaction between the extended -10 and `A/T`-rich UP-elements is one of the novel findings of Beagle that has not, to our knowledge, been reported in the literature previously.

In the case of *B. subtilis*, the `TG` extended -10 motif increases prediction accuracy only when accompanied by an UP-element. And in contrast to *E. coli*, use of the `TRTG` extended -10 increases prediction accuracy more than the `TG` extended -10. These differences between the species become clearer when we compare the sequence logos in Fig 2.

The best performing *E. coli* promoter model achieved 48% recall at 48% precision. In order to compare this result with other publications it is important to ensure that the experimental protocols are similar. In particular the prediction error tolerance and the length of upstream sequence being searched must be the same. We set up our experimental design to be similar to that of Huerta *et al.* (2003). Table 8e of their paper indicates a precision of 33% at a recall of 50%. For different experimental conditions, Burden *et al.* (2005) report 25% precision at 32% recall. When we modify our protocol to match theirs, we achieve 32% precision at 32% recall. The advantage of Beagle lies in the more complex promoter definition and in the iterative refinement of the PWMs. Different experimental conditions do not allow us to compare results with Gordon *et al.* (2006).

7 Conclusion

In this paper we introduced the software, *Beagle*, that enables the convenient description and exploration of PWM based promoter models. Beagle is a technically simple and fast method but nevertheless achieves state-of-the-art accuracy for TSS prediction.

Beagle has several additional attractive features. More complex promoter models can be constructed easily with an arbitrary number of PWMs and spacers. Training and prediction are fast, which allows an interactive study of promoter models and their elements. No negative examples are required for the training process, which can be a serious problem when building discriminative models such as SVMs. The generated models are completely transparent which is helpful for the testing of hypotheses.

We utilized Beagle to investigate a variety of models for σ^{70} promoters prediction in *E. coli* and *B. subtilis*. The results demonstrate an interesting interaction between UP-elements and extended -10 elements that has not been reported previously. The Beagle software, training and test data sets and extended results are publicly available at `http://eresearch.fit.qut.edu.au/Beagle/`.

Further work will examine the properties of wrongly predicted promoters. We also intend to apply Beagle to other transcription factors and genomes.

		E. coli		B. subtilis	
UP-element	extended -10	-	dist. GSS	-	dist. GSS
not used	-	37.5 ±1.4	43.3 ±1.2	61.6 ±1.8	61.2 ±1.7
	TG	36.1 ±1.4	41.6 ±1.3	59.4 ±1.8	62.5 ±1.8
	TRTG	32.5 ±1.3	37.6 ±1.3	59.2 ±1.8	62.6 ±1.8
proximal	-	39.0 ±1.3	44.3 ±1.4	65.2 ±1.9	66.4 ±2.0
	TG	35.4 ±1.3	43.7 ±1.3	66.2 ±2.1	68.5 ±2.1
	TRTG	31.5 ±1.2	38.6 ±1.3	67.3 ±1.9	70.3 ±1.9
full	-	34.8 ±1.3	41.4 ±1.2	58.8 ±1.7	62.0 ±1.7
	TG	31.4 ±1.3	39.0 ±1.4	64.8 ±1.6	66.7 ±1.8
	TRTG	25.9 ±1.0	35.4 ±1.2	65.0 ±1.8	66.6 ±1.9
A/T-rich	-	39.1 ±1.1	47.3 ±1.2	64.5 ±1.7	64.8 ±1.8
	TG	40.8 ±1.2	48.3 ±1.5	66.7 ±1.8	68.8 ±1.6
	TRTG	34.9 ±1.3	40.5 ±1.4	69.6 ±1.7	71.2 ±1.7

Table 1: Accuracies and 95% confidence intervals for TSS prediction on test data for different promoter models. Acceptance tolerance was ±5 bp. Averages are over 10-fold cross-validation, repeated 10 times.

References

Burden, S., Lin, Y.-X. & Zhang, R. (2005), 'Improving promoter prediction for the NNPP2.2 algorithm: a case study using *Escherichia coli* DNA sequences.', *Bioinformatics* **21**(5), 601–607. *http://dx.doi.org/10.1093/bioinformatics/bti047

Cartharius, K., Frech, K., Grote, K., Klocke, B., Haltmeier, M., Klingenhoff, A., Frisch, M., Bayerlein, M. & Werner, T. (2005), 'Matinspector and beyond: promoter analysis based on transcription factor binding sites.', *Bioinformatics* **21**(13), 2933–2942. *http://dx.doi.org/10.1093/bioinformatics/bti473

Estrem, S. T., Gaal, T., Ross, W. & Gourse, R. L. (1998), 'Identification of an UP element consensus sequence for bacterial promoters.', *Proc Natl Acad Sci U S A* **95**(17), 9761–9766.

Estrem, S. T., Ross, W., Gaal, T., Chen, Z. W., Niu, W., Ebright, R. H. & Gourse, R. L. (1999), 'Bacterial promoter architecture: subsite structure of UP elements and interactions with the carboxy-terminal domain of the RNA polymerase alpha subunit.', *Genes Dev* **13**(16), 2134–2147.

Frith, M. C., Li, M. C. & Weng, Z. (2003), 'Cluster-buster: Finding dense clusters of motifs in dna sequences.', *Nucleic Acids Res* **31**(13), 3666–3668.

Gordon, J. J., Towsey, M. W., Hogan, J. M., Mathews, S. A. & Timms, P. (2006), 'Improved prediction of bacterial transcription start sites.', *Bioinformatics* **22**(2), 142–148. *http://dx.doi.org/10.1093/bioinformatics/bti771

Gourse, R. L., Ross, W. & Gaal, T. (2000), 'UPs and downs in bacterial transcription initiation: The role of the alpha subunit of RNA polymerase in promoter recognition', *Mol Micro* **37**(4), 687–695.

Gross, C. A., Chan, C., Dombroski, A., Gruber, T., Sharp, M., Tupy, J. & Young, B. (1998), 'The functional and regulatory roles of sigma factors in transcription.', *Cold Spring Harb Symp Quant Biol* **63**, 141–155.

Grundy, W. N., Bailey, T. L., Elkan, C. P. & Baker, M. E. (1997), 'Meta-meme: motif-based hidden markov models of protein families.', *Comput Appl Biosci* **13**(4), 397–406.

Huerta, A. M. & Collado-Vides, J. (2003), 'Sigma70 promoters in *Escherichia coli*: specific transcription in dense regions of overlapping promoter-like signals.', *J Mol Biol* **333**(2), 261–278.

Miroslavova, N. S. & Busby, S. J. W. (2006), 'Investigation of the modular structure of bacterial promoters', *Biochem. Soc. Symp.* **73**, 1–10.

Mitchell, J. E., Zheng, D., Busby, S. J. W. & Minchin, S. D. (2003), 'Identification and analysis of 'extended -10' promoters in *Escherichia coli.*', *Nuc Acids Res* **31**(16), 4689–4695.

Murakami, K. S. & Darst, S. A. (2003), 'Bacterial RNA polymerases: the wholo story.', *Curr Opin Struct Biol* **13**(1), 31–9.

Schneider, T. D., Stormo, G. D., Gold, L. & Ehrenfeucht, A. (1986), 'Information content of binding sites on nucleotide sequences.', *J Mol Biol* **188**(3), 415–431.

Typas, A. & Hengge, R. (2005), 'Differential ability of sigma(s) and sigma70 of escherichia coli to utilize promoters containing half or full up-element sites.', *Mol Microbiol* **55**(1), 250–260. *http://dx.doi.org/10.1111/j.1365-2958.2004.04382.x

Vanet, A., Marsan, L. & Sagot, M. F. (1999), 'Promoter sequences and algorithmical methods for identifying them.', *Res Microbiol* **150**(9-10), 779–799.

Pathway to Functional Studies: Pipeline Linking Phylogenetic Footprinting and Transcription-Factor Binding Analysis

Nagesh Chakka **Jill E. Gready**

Computational Proteomics Group, John Curtin School of Medical Research, Australian National University,
Canberra, Australia. Email: nagesh.chakka@anu.edu.au; jill.gready@anu.edu.au

Abstract

Identification of transcription-factor binding sites is a critical first step in studying transcriptional regulation of genes. The comparative genomics method of phylogenetic footprinting is based on identifying sequence elements that are conserved across multiple genomes, and, thus, likely to be functional. We have developed a systematic high throughput screening pipeline to first search for conserved motifs using two different phylogenetic footprinting methods (motif-discovery and alignment-based) , and then rapid evaluate the motifs as potential transcription-factor binding sites. The results are displayed in an interactive graphical user interface, FactorScan, which integrates three separate complementary databases (conserved-sequence motifs, transcription-factor binding site motifs, TRANSFAC). We applied this pipeline for transcription-factor binding site analysis to the orthologous gene regions of prion-protein family genes from vertebrate lineages, taking account of the gene annotations.

Keywords: transcription factors; transcription factor binding sites; phylogenetic footprinting; TRANSFAC; MATCH; comparative genomics; sequence motifs.

1 Introduction

1.1 Motivation for the work

Availability of draft sequence for newly sequenced genomes of model organisms offers huge opportunities for characterizing functional elements using comparative genomic approaches. One key class of such functional elements is sites for binding proteins termed "transcription factors" (TFs) which play a central role in DNA polymerase II mediated transcriptional regulation of gene expression. TFs bind to specific short DNA sequence motifs know as TF binding sites (TFBSs) or *cis*-regulatory elements (CRE). Prediction of TFs which may bind to a particular gene can rapidly provide initial insights into potential functions of the target genes. This is based on known modes of actions of the TFs in regulating other better characterized genes. Such initial predictions can greatly assist in designing focused confirmatory experiments. As TFBSs are under greater selective pressure than other non-protein-coding DNA, the reliability of predicting them is greatly improved by comparative genomics to filter out noise from genetic drift. Identifying such conserved sequence elements in non-coding regions of homologous genes from phylogenetic comparison is called 'phylogenetic footprinting'(PF) (Tagle, Koop, Goodman, Slightom, Hess & Jones 1988). While there are several online resources which can perform PF, none provides the flexibility for combining the conserved sequence-motif data with TFBS analysis and, at the same time, allowing the flexibility to customize the searches based on gene annotation information. To address this deficiency, we developed a two-step procedure which combines PF with TFBS analysis. This automated pipeline enables us to carry out rapid screening and evaluation of the phylogenetically conserved motifs for potential TF-binding affinity. To perform the most comprehensive searches, TRANSFAC professional database (version 9.2) was included in the pipeline. We used this strategy to identify potential TFBSs in prion protein and its paralogous gene, doppel, encoded by the *PRNP* and *PRND* genes, respectively. We gleaned some initial insights into the functions of these genes, which are not well understood, from the TFs predicted to be involved in regulating their expression.

1.2 Advantages of our approach

As TFBSs are short DNA motifs of 5-15 bp, analyzing a single sequence would lead to a very high percentage of false positive hits. PF offers a solution to this problem by identifying such sequence elements that are conserved among genes that are either orthologous or co-expressed. Several programs implement PF but only a few combine it with TFBS analysis, for example, rVISTA (Loots, Ovcharenko, Pachter, Dubchak & Rubin 2002) and ConSite (Sandelin, Wasserman & Lenhard 2004). However, both programs allow only pairwise comparison; there are no programs which perform this analysis on multiple sequences. Another restriction with rVISTA and ConSite is that they use different databases of position weight matrices (PWMs), TRANSFAC public and JASPAR respectively, neither of which is as comprehensive as TRANSFAC professional. Finally, rVISTA and ConSite do not provide a facility to customize display of the results to make the maximum use of the output, for example, display of clusters of TFs. Our approach overcomes these restrictions, by providing various options for customizing searches for both pairwise and multiple sequences, for incorporating flexibility in visualizing the output, and for using databases of PWMs of choice.

2 Pipeline for Phylogenetic Footprinting (PF) Analysis

2.1 Rationale for selecting algorithms

For alignment-based identification of conserved elements, we used AVID (Bray, Dubchak & Pachter 2003) and LAGAN (Brudno, Do, Cooper, Kim, Davydov, Program, Green, Sidow & Batzoglou 2003), both of which are sensitive and widely used for genome-wide alignment problems. rVISTA uses both alignment programs and LAGAN is being incorporated in ConSite alignment step. For identification of conserved elements from multiple sequences, we used FootPrinter (Blanchette & Tompa 2003) which takes phylogeny into account and, hence, weighs the sequence based on the evolutionary relationship and implements most of the concepts of PF, in contrast to other motif-discovery methods such as MEME (Bailey & Elkan 1994). BioProspector (Liu, Brutlag & Liu 2001) identifies motifs that are overrepresented in the input sequences and, hence, is a different approach to handling this problem.

2.2 Annotated gene sequence database

A database of annotated gene sequences was created by mapping the (*PRNP* and *PRND*) cDNA sequence obtained from either experiments or public databases onto the genome sequence obtained from various genome sequencing projects. The EMBOSS application (Rice, Longden & Bleasby 2000) "est2genome" was used to annotate the exon-intron boundaries, transcription start site, while "getorf" was used for detecting the coding regions, which were then masked. Genomic sequence covering 2 kb upstream to the transcription start site, the whole of exon-intron region, and 2 kb downstream from the transcription stop site was included in the PF analysis. To improve the signal-to-noise ratio, we selected representative species for which genomic data for *PRNP* and *PRND* was available. This comprises several eutherian mammalian species, and all those available for lower vertebrates; marsupial mammals *Monodelphis domestica* (South American opossum) and Tammar wallaby, chicken, and the frog *Xenopus tropicalis.* Indicative sequence lengths are shown in the scale bar of Figure 5 (b) for the complete genomic regions of mouse and human *PRNP* and *PRND*; there are significant differences in the lengths of the intronic and intergenic regions of these genes, both among eutherian mammals and among the vertebrate lineages due to the high frequency of insertion of transposable elements (Premzl, Gready, Jermiin, Simonic & Graves 2004).

2.3 Conserved-sequence motif detection

Conserved sequence motifs were identified by several PF methods which we categorize into two groups, alignment-based and motif-discovery-based. Separate pipelines for each, alignment-based (Fig. 1(a)) and motif discovery-based (Fig. 1(b)), were developed.

2.3.1 Alignment-based method.

To perform end-to-end comparisons, the global pairwise-alignment methods AVID (Bray et al. 2003) and LAGAN (Brudno et al. 2003) were used independently to generate pairwise alignments. The AVID alignment method is fast, memory efficient, and practical for sequence alignments of large genomic regions up to megabase. AVID performs the pairwise alignment of two input sequences; the output comprises the alignment and additional information. The alignment files were used for downstream processing. LAGAN is a method for rapid global alignment of two homologous sequences. The algorithm is based on three main steps (Brudno et al. 2003): (1) generation of pairwise local alignments, (2) construction of a rough global map, by linking a subset of local alignments, and (3) computation of the final global alignment. LAGAN alignments were generated using the translate anchor option and binary output format was selected, which enables downstream processing. Both the AVID and LAGAN alignments for all possible pairwise combinations (Fig. 2) of sequences in the annotated gene sequence database were performed using the Perl script "doAlign.pl".

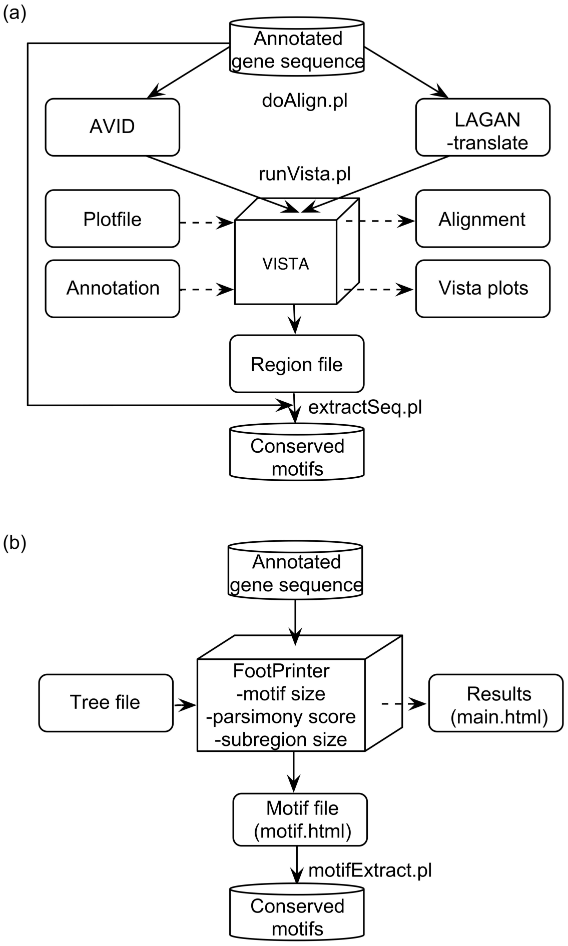


Figure 1: Phylogenetic footprinting pipeline using AVID/LAGAN alignment methods and annotation with VISTA (a). Pipeline summarizing steps in phylogenetic footprinting using FootPrinter (b). The end result of both analyses is a database of the conserved sequence motifs.

Annotation with VISTA. Global pairwise alignments generated by AVID and LAGAN were annotated using VISTA (Frazer, Pachter, Poliakov, Rubin & Dubchak 2004). VISTA can be configured by changing several parameters (e.g. percentage identity and length), which can be defined in the input Plotfile. To facilitate trialing of several combinations of percent identity (range: 75% to 100%) and length (range: 8 to 15 bp) values, a Perl script "runVista.pl" was developed to generate corresponding Plotfiles for percent identity and length values passed as command line arguments. VISTA generates three output files: VISTA plot, alignment, and region file (Fig. 1(a)).

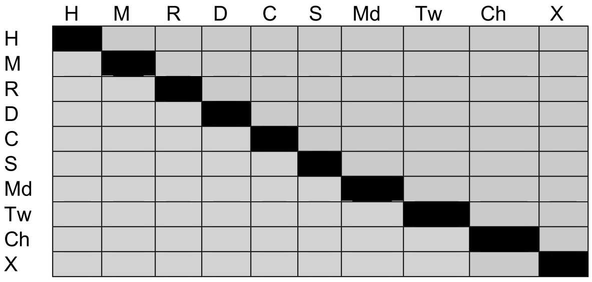


Figure 2: Summary of pairwise sequence comparisons (all grey cells) performed with AVID and LAGAN between the species on the X and Y axes. H, human; M, mouse; R, rat; D, dog; C, cow; S, sheep; Md, *Monodelphis*; Tw, Tammar wallaby; Ch, Chicken; X, *Xenopus.*

VISTA plot contains graphical representation of the conserved regions. The region file contains details of those regions which satisfied the user-specified length and percentage cutoffs. This file was processed using a Perl script "extractseq.pl". Based on the start and end numbers of the conserved regions, the subsequences were extracted using the EMBOSS application "extractseq" integrated into "extractseq.pl". This process was repeated for all the region files obtained for the various combinations of alignments. Finally, the Perl script generates a single multi-FASTA file of all the conserved subsequences, which are stored in a conserved sequence database. The sequence identifier for each motif contains information about the pair involved in the alignment, its position in both the reference sequences and the region to which it belongs. This enables the exact position of the conserved sequence to be tracked for further analysis. For those motifs which are shorter than 15 bp, continuous stretches of five "N" were added to both the 5' and 3' ends of the motif to facilitate the TFBS analysis.

2.3.2 Motif-discovery approach.

FootPrinter (Blanchette & Tompa 2003) implements motif-discovery method to identify conserved motifs in a collection of homologous sequences. The algorithm identifies each set of motifs of user-defined size, one from each input sequence, that have a parsimony score specified by the user. This process uses phylogenetic tree information. The input for FootPrinter is the file containing sequences from the annotated gene sequence database and a tree file (Fig. 1b). The program generates several output files with different file formats. For programmatic processing, html output format was selected. The input sequences were divided into several datasets: intra-eutherian mammals and others comprising sets with eutherian mammals and sequences from one or more of the other lineages (marsupial, avian, amphibian). The output motif file (motif.html) contains the information about the motif and its position. A comprehensive search was performed using different FootPrinter options (subregion- 1000 to 3000bp; motif size- 6 to 10bp; parsimony score- 0 to 2). Using a Perl script "motifextract.pl", the "motif.html" output file was converted to a single multi-FASTA file. Each analysis was performed twice using upstream and downstream (FootPrinter: sequence_type) option. The multi-FASTA files from both analyses were combined using a Perl script, "compileTFBS.pl" to produce a non-redundant single multi-FASTA file. These multi-FASTA files relating to different subregion sizes were stored in a conserved-sequence database. Each sequence-motif position was registered in the sequence identifier.

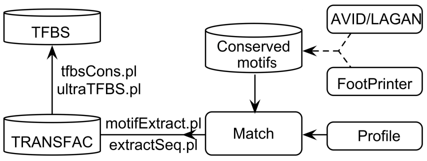
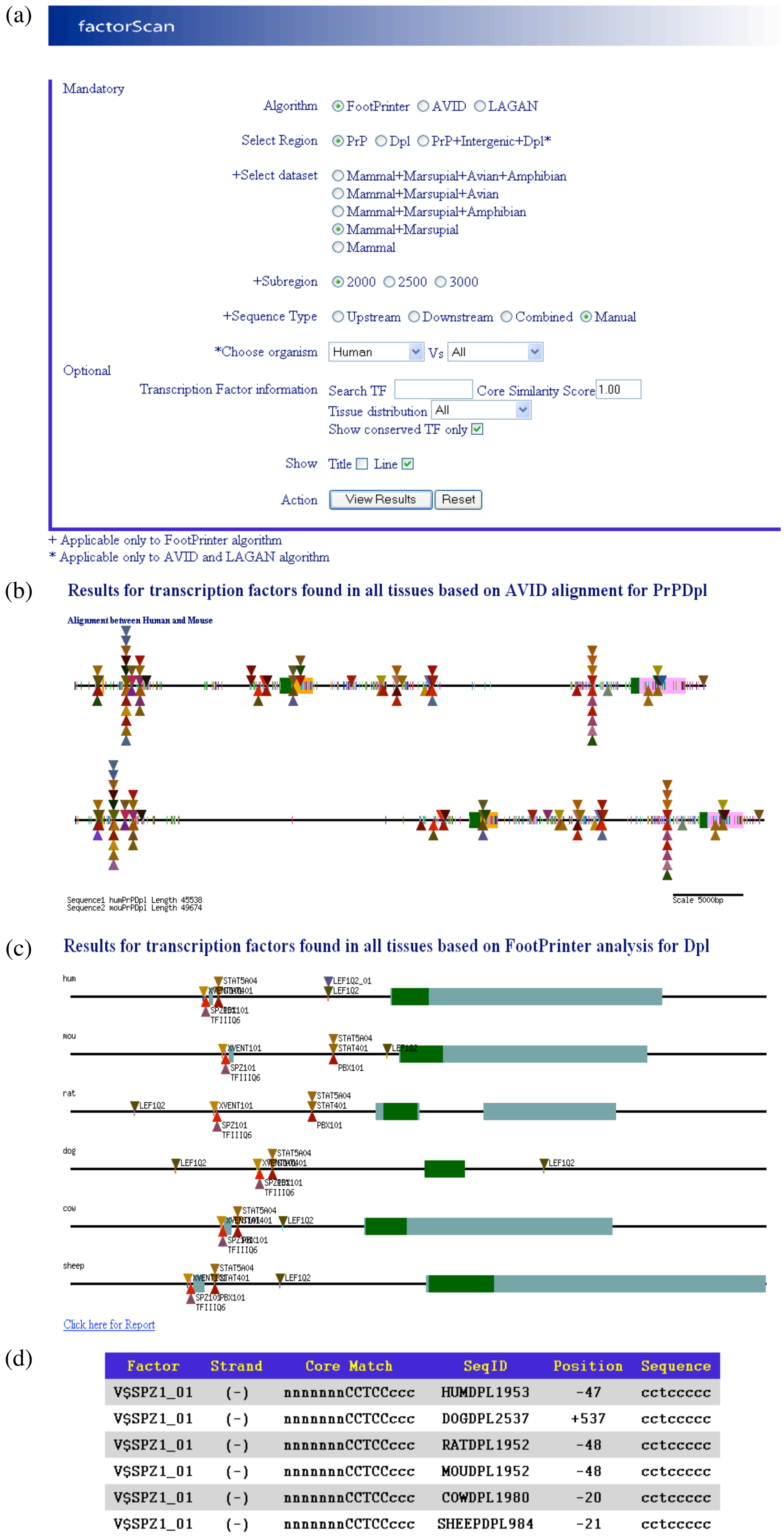


Figure 3: Pipeline showing the steps in the TFBS analysis.

3 Pipeline for transcription-factor binding-site (TFBS) analysis

To enable a comprehensive analysis, the commercial version TRANSFAC (Matys, Fricke et al. 2003) professional was used for TFBS analysis. MATCH (Kel, Gossling et al. 2003) is a tool which uses the weight matrices in the TRANSFAC database to search for putative TFBSs; the advanced version, MATCH professional, distributed with TRANSFAC professional was used. Published TFBS information was used to optimize the MATCH search parameters, i.e. to predict maximum true positives and minimum false positives against the TRANSFAC professional database. A systematic pipeline was developed to assess the specificity of TF binding to the conserved-sequence motifs identified by phylogenetic footprinting (Fig. 3). The steps of the analysis were:

- Starting inputs were the motifs identified by AVID/LAGAN/FootPrinter methods.
- These motifs were scored against the TRANSFAC database using MATCH which uses the information defined in the profile (selection of matrices with defined cutoffs).
- The output file generated by MATCH was processed to eliminate entries for motif sequences which did not correlate with any known binding affinity; only sequences showing putative binding to the vertebrate TFs were retained.
- The Perl scripts, "motifExtract.pl" and "extractSeq.pl" contain modules that process the MATCH output file.
- The final output (same format as MATCH output) generated by these Perl scripts was stored in the TFBS database.
- When conserved motifs were obtained by non-stringent criteria, e.g. for parsimony score value > 0 for FootPrinter or percent identity value $<$ 100% for alignment methods, it is possible that TFs predicted to bind to the same set of conserved motifs in different input sequences could differ. Such predicted TFs were eliminated. This criterion was implemented by two Perl scripts, "tfbsCons.pl" and "ultraTFBS.pl" which need to be run consecutively.
- Altogether, the resultant predicted motifs were classified as either highly conserved or less highly conserved. Both sets were stored in the TFBS database.



(a)

factorScan

Mandatory

Algorithm ● FootPrinter ○ AVID ○ LAGAN

Select Region ● PrP ○ Dpl ○ PrP+Intergenic+Dpl*

+Select dataset
○ Mammal+Marsupial+Avian+Amphibian
○ Mammal+Marsupial+Avian
○ Mammal+Marsupial+Amphibian
● Mammal+Marsupial
○ Mammal

+Subregion ● 2000 ○ 2500 ○ 3000

+Sequence Type ○ Upstream ○ Downstream ○ Combined ● Manual

*Choose organism Human Vs All

Optional

Transcription Factor information Search TF Core Similarity Score 1.00
Tissue distribution All
Show conserved TF only ☑

Show Title □ Line ☑

Action View Results Reset

+ Applicable only to FootPrinter algorithm
* Applicable only to AVID and LAGAN algorithm

(b) Results for transcription factors found in all tissues based on AVID alignment for PrPDpl

Alignment between Human and Mouse
Sequence1 humPrPDpl Length 45538
Sequence2 mouPrPDpl Length 49674
Scale 5000bp

(c) Results for transcription factors found in all tissues based on FootPrinter analysis for Dpl

Click here for Report

(d)

Factor	Strand	Core Match	SeqID	Position	Sequence
V$SPZ1_01	(-)	nnnnnnnCCTCCccc	HUMDPL1953	-47	cctccccc
V$SPZ1_01	(-)	nnnnnnnCCTCCccc	DOGDPL2537	+537	cctccccc
V$SPZ1_01	(-)	nnnnnnnCCTCCccc	RATDPL1952	-48	cctccccc
V$SPZ1_01	(-)	nnnnnnnCCTCCccc	MOUDPL1952	-48	cctccccc
V$SPZ1_01	(-)	nnnnnnnCCTCCccc	COWDPL1980	-20	cctccccc
V$SPZ1_01	(-)	nnnnnnnCCTCCccc	SHEEPDPL984	-21	cctccccc

Figure 5: (a) Web form for the user to submit information required for viewing the results. (b) Results page for alignment method and (c) for FootPrinter method. Note the different options in the display pattern shown in (b) and (c); TF titles are seen in (c), while conserved-sequence motifs are seen as vertical bars in (b). (d) Report page showing the report for the search made for the TF SpZ1.

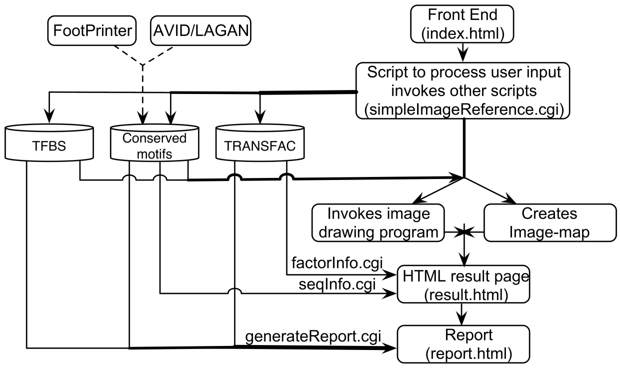


Figure 4: The procedural flow of information, starting from submitting the web form to the display of results, and the programs involved with each task.

4 Visual front-end for data analysis

TFBSs occur in combinations of order, distance and strand orientation which are specific for a particular gene. Analyzing this organization in relation to the gene structure is essential for understanding transcriptional regulation. An intuitive visual front-end is necessary to allow the researcher to view the TFBS organization and interpret and evaluate the results. To achieve this, we designed an interactive user-interface, FactorScan.

4.1 Interface development

FactorScan is a web-based application accessible through a web browser. It links the TFBS information, conserved-sequence motif information predicted by AVID/LAGAN/FootPrinter and the TRANSFAC database (Fig. 4). This interface enables access to the data (conserved-sequence motifs and TFBS) generated by the various pipelines (Figs 1 and 3): it is not dynamically generated during the visualisation. The web interface has three main components, the web form, the results page and the report page.

Input: The user input for the web form is categorized into mandatory and optional parameters (Fig 5(a)). The mandatory parameters include the gene for which the results are to be displayed and the various options used for phylogenetic footprinting to generate the data (subregion size, sequence type, sequence dataset). The optional parameters are for customizing and controlling the display of the results. Some important features are (i) Transcription Factor Search, (ii) Core Similarity Score, (iii) Title, (iv) Tissue Source and (v) Line. The Transcription Factor Search is useful to display a subset of TFs of particular interest, either individually or in combinations. The latter is particularly useful for identifying and comparing 'modules' TFBS (clusters of TFBS in a defined order)(Wasserman and Sandelin 2004). The Core Similarity Score can be used to visualize TFs which satisfy criteria set by the user. This value is in the range of 0-1; by default this is set to 1 to display the statistically most significant hits. The "Title" option can be used to visualize the name of the TF matrices for the displayed TFs. Tissue-specific TFs can be searched according to tissue, such as brain and testis. The cell-positive and cell-negative information in the TRANSFAC database is used for this purpose. The conserved-sequence motif distribution can be viewed by selecting the "Line" option.

Output: The submitted web form is processed by a CGI script "simpleImageReference.cgi" (Fig. 4) and the results are displayed in the same window. The results page displays a schematic of relative organization of gene annotation, TF and conserved-sequence motif information. Genomic sequence is represented, conventionally, as a horizontal line with exons mapped on as rectangular boxes, and with coding- and non-coding regions of exons shaded in different colors (Fig. 5(b),(c)). The TFs predicted to bind are represented as triangles (Fig. 5(b), (c)), inverted and upright for the forward and reverse strands, respectively. Each TF is assigned a unique color; its name is displayed if the "Title" option is selected. The conserved-sequence motifs, identified by any of the methods, are represented as vertical bars (Fig. 5(b)); use of color is particularly helpful to discriminate these regions when they are very close. Triangles representing TFs and vertical bars representing conserved-sequence motifs are clickable areas. Clicking on the triangle displays a summary of TF information, obtained from the TRANSFAC database. Clicking on the vertical bar displays information about the conserved-sequence motif, accessed from information in the conserved-sequence motif database. This is particularly useful as the conserved-sequence motif can be examined for other purposes. For Footprinter analyses the schematic is drawn to scale within a species, but between species the scale is not normalized (Fig. 5(c)). For pairwise-alignment analyses, the scale (also shown; see Fig. 5(c)) is normalized between the pairs, and the results can be displayed either between specific pairs or for one against all others. The latter is useful to compare the conserved TFBS distribution among various lineages. Information about species, abbreviations used and the sequence length in base pairs is provided in table form at the bottom of the schematic. The results page also has a link to view the report of the TFs and their binding sites. Clicking this link pops up a window (Fig. 5(d)) displaying a detailed summary of the TFs, the strand to which it binds, core match, the conserved-sequence motif identifier, the position of the TF relative to the transcription start site and the sequence which was used for TFBS analysis.

5 Analysis of results

Our use of this combinatorial PF approach (i.e. both alignment-based and motif-discovery-based methods) predicted most of the known TFs for the *PRNP* and *PRND* genes. The SP1 TF has been shown experimentally to play a role in transcriptional regulation of *PRNP* (Saeki, Matsumoto, Matsumoto & Onodera 1996)(Baybutt & Manson 1997)(Inoue, Tanaka, Horiuchi, Ishiguro & Shinagawa 1997)(Mahal, Asante, Antoniou & Collinge 2001). Mahal and coworkers also found AP1 and AP2 binding sites in the human promoter region. We predicted both SP1 and AP1/AP2 TFBSs using the pairwise-alignment method in most pairs of sequences compared, but these TFs were not identified using FootPrinter analysis (motif absence in any sequence was not allowed). Premzl and coworkers (Premzl, Delbridge, Gready, Wilson, Johnson, Davis, Kuczek & Graves 2005) reported several regulatory regions in *PRNP* using PF (Footprinter method) with the then-available sequences (eutherian mammals and one marsupial only): most of the TFs (MEF2, Oct-1, MyT1 and NFAT) were predicted in the intra-eutherian mammal comparison.

Nagyova and coworkers (Nagyov, Pastorek & Kopcek 2004) experimentally validated the role of USF and NF-Y in *PRND* promoter activity. We predicted the NF-Y region using both alignment-based and FootPrinter methods. We predicted the USF-binding site in comparisons of some sequence pairs using alignment-based methods but not using FootPrinter: this indicates either that the USF-binding site is degenerate or short or that it is not phylogenetically conserved among the species compared.

Of particular interest for our genes, we predicted several new TFBSs (*PRNP*: E4BP4, DBP, FAC1, MYB; *PRND*: SpZ1, CDXA, LEF1) which are phylogenetically conserved for both genes, and which correlate well with physiological behaviour consistent with operation of these TFs in regulating other genes (e.g. tissue specificity, specific physiological role). We are testing these predictions experimentally, and so far have indicative confirmation for FAC1 and SpZ1.

6 Application and Conclusions

We have developed a graphical web interface to facilitate researcher evaluation of results from the phylogenetic footprinting and TFBS analysis pipelines. An application of the pipeline and web interface is illustrated by an analysis on *PRNP* and *PRND* genes. This revealed several new conserved TFBSs, in addition to detecting already published and experimentally validated TFs for regulating these genes. Detection of the latter serves as a confidence test for our pipeline analysis. Several of the newly predicted TFBSs are consistent with the known functions of these genes, providing strong starting points for follow up experimental studies. A combinatorial approach of predicting conserved motifs using FootPrinter and AVID/LAGAN methods followed by TF binding analysis significantly improved the confidence in the predicted TFBSs. Our pipeline was also tested on the newly discovered prion-protein family gene, SPRN, coding for the protein Shadoo, providing us with valuable initial functional predictions of a gene whose function is not known. Our development of a pipeline which incorporates both alignment-based and motif-discovery based methods with TFBS analysis is novel, and provides a powerful new tool for high throughput, robust analysis. The concurrent development of the graphical-display module to this pipeline, greatly enhances its usefulness by facilitating intuitive and interactive analysis of the results.

7 Software and Hardware

Standalone versions of AVID (version 2.1), LAGAN (version 1.21) and FootPrinter (version 2.1) were used for the phylogenetic footprinting analysis. The TRANSFAC database version 9.2 and MATCH version 6.1 were used for TFBS analysis. The web form was implemented using HTML running on an Apache web server on a Linux operating system at valera.anu.edu.au which hosts the web page and can be accessed locally with the web address http://valera.anu.edu.au:8080/factorScan_html. The graphical package Perl GD and Common Gateway Interface package Perl CGI were used for the web interface development. Additional pipelining and analysis modules were written in Perl. All analysis was performed on a PC but some of the more memory-demanding FootPrinter analyses were performed on the Dell Linux cluster at the APAC (Australian Partnership for Advanced Computing) National Facility.

References

Bailey, T. L. & Elkan, C. (1994), ‘Fitting a mixture model by expectation maximization to discover motifs in biopolymers.’, *Proc Int Conf Intell Syst Mol Biol* **2**, 28–36.

Baybutt, H. & Manson, J. (1997), ‘Characterisation of two promoters for prion protein (prp) gene expression in neuronal cells.’, *Gene* **184**(1), 125–131.

Blanchette, M. & Tompa, M. (2003), ‘Footprinter: A program designed for phylogenetic footprinting.’, *Nucleic Acids Res* **31**(13), 3840–3842.

Bray, N., Dubchak, I. & Pachter, L. (2003), ‘Avid: A global alignment program.’, *Genome Res* **13**(1), 97–102.

Brudno, M., Do, C. B., Cooper, G. M., Kim, M. F., Davydov, E., Program, N. I. S. C. C. S., Green, E. D., Sidow, A. & Batzoglou, S. (2003), ‘Lagan and multi-lagan: efficient tools for large-scale multiple alignment of genomic dna.’, *Genome Res* **13**(4), 721–731.

Frazer, K. A., Pachter, L., Poliakov, A., Rubin, E. M. & Dubchak, I. (2004), ‘Vista: computational tools for comparative genomics.’, *Nucleic Acids Res* **32**(Web Server issue), W273–W279.

Inoue, S., Tanaka, M., Horiuchi, M., Ishiguro, N. & Shinagawa, M. (1997), ‘Characterization of the bovine prion protein gene: the expression requires interaction between the promoter and intron.’, *J Vet Med Sci* **59**(3), 175–183.

Kel, A. E., Gssling, E., Reuter, I., Cheremushkin, E., Kel-Margoulis, O. V. & Wingender, E. (2003), ‘Match: A tool for searching transcription factor binding sites in dna sequences.’, *Nucleic Acids Res* **31**(13), 3576–3579.

Liu, X., Brutlag, D. L. & Liu, J. S. (2001), ‘Bioprospector: discovering conserved dna motifs in upstream regulatory regions of co-expressed genes.’, *Pac Symp Biocomput* pp. 127–138.

Loots, G. G., Ovcharenko, I., Pachter, L., Dubchak, I. & Rubin, E. M. (2002), ‘rvista for comparative sequence-based discovery of functional transcription factor binding sites.’, *Genome Res* **12**(5), 832–839.

Mahal, S. P., Asante, E. A., Antoniou, M. & Collinge, J. (2001), ‘Isolation and functional characterisation of the promoter region of the human prion protein gene.’, *Gene* **268**(1-2), 105–114.

Matys, V., Fricke, E., Geffers, R., Gssling, E., Haubrock, M., Hehl, R., Hornischer, K., Karas, D., Kel, A. E., Kel-Margoulis, O. V., Kloos, D.-U., Land, S., Lewicki-Potapov, B., Michael, H., Mnch, R., Reuter, I., Rotert, S., Saxel, H., Scheer, M., Thiele, S. & Wingender, E. (2003), ‘Transfac: transcriptional regulation, from patterns to profiles.’, *Nucleic Acids Res* **31**(1), 374–378.

Nagyov, J., Pastorek, J. & Kopcek, J. (2004), ‘Identification of the critical cis-acting elements in the promoter of the mouse prnd gene coding for doppel protein.’, *Biochim Biophys Acta* **1679**(3), 288–293.

Premzl, M., Delbridge, M., Gready, J. E., Wilson, P., Johnson, M., Davis, J., Kuczek, E. & Graves, J. A. M. (2005), ‘The prion protein gene: identifying regulatory signals using marsupial sequence.’, *Gene* **349**, 121–134.

Premzl, M., Gready, J. E., Jermiin, L. S., Simonic, T. & Graves, J. A. M. (2004), 'Evolution of vertebrate genes related to prion and shadoo proteins–clues from comparative genomic analysis.', *Mol Biol Evol* **21**(12), 2210–2231.

Rice, P., Longden, I. & Bleasby, A. (2000), 'Emboss: the european molecular biology open software suite.', *Trends Genet* **16**(6), 276–277.

Saeki, K., Matsumoto, Y., Matsumoto, Y. & Onodera, T. (1996), 'Identification of a promoter region in the rat prion protein gene.', *Biochem Biophys Res Commun* **219**(1), 47–52.

Sandelin, A., Wasserman, W. W. & Lenhard, B. (2004), 'Consite: web-based prediction of regulatory elements using cross-species comparison.', *Nucleic Acids Res* **32**(Web Server issue), W249–W252.

Tagle, D. A., Koop, B. F., Goodman, M., Slightom, J. L., Hess, D. L. & Jones, R. T. (1988), 'Embryonic epsilon and gamma globin genes of a prosimian primate (galago crassicaudatus). nucleotide and amino acid sequences, developmental regulation and phylogenetic footprints.', *J Mol Biol* **203**(2), 439–455.

Wasserman, W. W., Sandelin, A. (2004), 'Applied bioinformatics for the identification of regulatory elements.', *Nat Rev Genet* **5(4)**, 276–87.

Modified global k-means algorithm for clustering in gene expression data sets

Adil M. Bagirov **Karim Mardaneh**

Centre for Informatics and Applied Optimization,
School of Information Technology and Mathematical Sciences,
University of Ballarat, Victoria, 3353, Australia,
Email: a.bagirov@ballarat.edu.au

Abstract

Clustering in gene expression data sets is a challenging problem. Different algorithms for clustering of genes have been proposed. However due to the large number of genes only a few algorithms can be applied for the clustering of samples. k-means algorithm and its different variations are among those algorithms. But these algorithms in general can converge only to local minima and these local minima are significantly different from global solutions as the number of clusters increases. Over the last several years different approaches have been proposed to improve global search properties of k-means algorithm and its performance on large data sets. One of them is the global k-means algorithm. In this paper we develop a new version of the global k-means algorithm: the modified global k-means algorithm which is effective for solving clustering problems in gene expression data sets. We present preliminary computational results using gene expression data sets which demonstrate that the modified k-means algorithm improves and sometimes significantly results by k-means and global k-means algorithms.

1 Introduction

This paper develops an incremental algorithm for solving sum-of-squares clustering problems in gene expression data sets. Clustering in gene expression data sets is a challenging problem. Different algorithms for clustering of genes have been proposed (see, for example, (Medvedovic & Sivaganesan 2002, Yeung et al. 2001, Yeung et al. 2003)). However due to the large number of genes only a few algorithms can be applied for the clustering of samples ((Bagirov et al. 2003)). As the number of clusters increases the number of variables in the clustering problem increases drastically and most of clustering algorithms become inefficient for solving such problems. k-means algorithm and its different variations are among those algorithms which still applicable to clustering of samples in gene expression data sets. But k-means algorithms in general can converge only to local minima and these local minima may be significantly different from global solutions as the number of clusters increases. Recently the global k-means algorithm has been proposed to improve global search properties of k-means algorithms ((Likas et al. 2003)). In this paper we develop a new version of the global k-means algorithm: the modified global k-means algorithm which is effective for solving clustering problems in gene expression data sets.

The cluster analysis deals with the problems of organization of a collection of patterns into clusters based on similarity. It is also known as the *unsupervised* classification of patterns and has found many applications in different areas. In cluster analysis we assume that we have been given a finite set of points A in the n-dimensional space $\mathbb{R}^n$, that is

$$A = \{a^1, \ldots, a^m\}, \text{ where } a^i \in \mathbb{R}^n, \ i = 1, \ldots, m.$$

There are different types of clustering. In this paper we consider the hard unconstrained partition clustering problem, that is the distribution of the points of the set A into a given number k of disjoint subsets $A^j, \ j = 1, \ldots, k$ with respect to predefined criteria such that:

1) $A^j \neq \emptyset, \ j = 1, \ldots, k;$

2) $A^j \bigcap A^l = \emptyset, \ j, l = 1, \ldots, k, \ j \neq l;$

3) $A = \bigcup\limits_{j=1}^{k} A^j.$

4) no constraints are imposed on clusters $A^j, \ j = 1, \ldots, k.$

The sets $A^j, \ j = 1, \ldots, k$ are called clusters. We assume that each cluster A^j can be identified by its center (or centroid) $x^j \in \mathbb{R}^n, \ j = 1, \ldots, k$. Then the clustering problem can be reduced to the following optimization problem (see (Bock 1998, Spath 1980)):

$$\text{minimize } \ \psi(x, w) = \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{k} w_{ij} \|x^j - a^i\|^2 \quad (1)$$

subject to

$$x = (x^1, \ldots, x^k) \in \mathbb{R}^{n \times k}, \quad (2)$$

$$\sum_{j=1}^{k} w_{ij} = 1, \ i = 1, \ldots, m, \quad (3)$$

and

$$w_{ij} = 0 \text{ or } 1, \ i = 1, \ldots, m, \ j = 1, \ldots, k \quad (4)$$

where w_{ij} is the association weight of pattern a^i with cluster j, given by

$$w_{ij} = \begin{cases} 1 & \text{if pattern } a^i \text{ is allocated to cluster } j, \\ 0 & \text{otherwise} \end{cases}$$

and

$$x^j = \frac{\sum_{i=1}^m w_{ij} a^i}{\sum_{i=1}^m w_{ij}}, \quad j = 1, \ldots, k.$$

Here $\|\cdot\|$ is an Euclidean norm and w is an $m \times k$ matrix. The problem (1)-(4) is also known as minimum sum-of-squares clustering problem.

Different algorithms have been proposed to solve the clustering problem. The paper (Jain et al. 1999) provides survey of most of existing algorithms. We mention among them heuristics like k-means algorithms and their variations (h-means, j-means etc.), mathematical programming techniques including dynamic programming, branch and bound, cutting plane, interior point methods, the variable neighborhood search algorithm and metaheuristics like simulated annealing, tabu search, genetic algorithms (see (Al-Sultan 1995, Brown & Entail 1992, de Merle et al. 2001, Diehr 1985, Dubes & Jain 1976, Hanjoul & Peeters 1985, Hansen & Jaumard 1997, Hansen & Mladenovic 2001a, Hansen & Mladenovic 2001b, Koontz et al. 1975, Selim & Al-Sultan 1991, Spath 1980, Sun et al. 1994)). Since the number of genes in gene expression data sets are very large most of these algorithms cannot be applied for clustering of samples in such data sets.

The problem (1)-(4) is a global optimization problem and the objective function ψ in this problem has many local minima. However clustering algorithms based on global optimization techniques are not applicable to even relatively large data sets. Algorithms which are applicable to such data sets can locate only local minima of the function ψ and these local minima can differ from global solutions significantly as the number of clusters increases. Another difficulty is that the number of clusters, as a rule, is not known a priori. Over the last several years different incremental algorithms have been proposed to address these difficulties. Results of numerical experiments show that an incremental approach allows one, as a rule, to locate a local solution close to global one. Consequently it can produce a better cluster structure of a data set. The paper (Bagirov & Yearwood, 2006) develops an incremental algorithm based on nonsmooth optimization approach to clustering. The global k-means algorithm was developed in (Likas et al. 2003). The incremental approach is also discussed in (Hansen et al. 2004).

In this paper we propose a new version of the global k-means algorithm for solving clustering problems in gene expression data sets. In this algorithm a starting point for the k-th cluster center is computed by minimizing so-called auxiliary cluster function. We present the results of numerical experiments with 6 gene expression data sets. These results demonstrate that the proposed algorithm improves solutions obtained by the global k-means algorithm and for some data sets this improvement is substantial.

The rest part of the paper is organized as follows: Section 2 gives a brief description of k-means and the global k-means algorithms. The nonsmooth optimization approach to clustering and an algorithm for the computation of a starting point is described in Section 3. Section 4 presents an algorithm for solving clustering problems. The results of numerical experiments are given in Section 5 and Section 6 concludes the paper.

2 k-means and the global k-means algorithms

In this section we give a brief description of k-means and the global k-means algorithms.

The k-means algorithm proceeds as follows:

1. choose a seed solution consisting of k centers (not necessarily belonging to A);
2. allocate data points $a^i \in A$ to its closest center and obtain k-partition of A;
3. recompute centers for this new partition and go to Step 2 until no more data points change cluster.

The effectiveness of this algorithm highly depends on a starting point. It converges only to a local solution which can significantly differ from the global solution in many large data sets.

The global k-means algorithm proposed in (Likas et al. 2003) computes clusters successively. At the first iteration of this algorithm the centroid of the set A is computed and in order to compute k-partition at the k-th iteration this algorithm uses centers of $k-1$ clusters from the previous iteration. The global k-means algorithm for the computation of $q \le m$ clusters in a data set A can be described as follows.

Algorithm 1 The global k-means algorithm.

Step 1. (Initialization) Compute the centroid x^1 of the set A:

$$x^1 = \frac{1}{m} \sum_{i=1}^m a^i, \quad a^i \in A, \ i = 1, \ldots, m$$

and set $k = 1$.

Step 2. Set $k = k + 1$ and consider the centers $x^1, x^2, \ldots, x^{k-1}$ from the previous iteration.

Step 3. Consider each point a of A as a starting point for the k-th cluster center, thus obtaining m initial solutions with k points $(x^1, \ldots, x^{k-1}, a)$; apply k-means algorithm to each of them; keep the best k-partition obtained and its centers $x^1, x^2, \ldots, x^k$.

Step 4. (Stopping criterion) If $k = q$ then stop, otherwise go to Step 2.

This version of the algorithm is not applicable for clustering on middle sized and large data sets. Two procedures were introduced to reduce its complexity (see (Likas et al. 2003)). We mention here only one of them because the second procedure is applicable to low dimensional data sets. Let d^i_{k-1} be a squared distance between $a^i \in A$ and the closest cluster center among the $k-1$ cluster centers obtained so far. For each $a^i \in A$ we calculate the following:

$$r_i = \sum_{j=1}^m \min\{0, \|a^i - a^j\|^2 - d^j_{k-1}\}$$

and we take the data point $a^l \in A$ for which

$$l = \arg\min{}_{i=1,\ldots,m} r_i$$

as a starting point for the k-th cluster center. Then k-means algorithm is applied starting from the point $x^1, x^2, \ldots, x^{k-1}, a^l$ to find k cluster centers. In our numerical experiments we use this procedure.

It should be noted that k-means algorithm and its variants tend to produce only spherical clusters and they are not always appropriate for solving clustering problems. However applying k-means algorithms we assume that clusters in a data set can be approximated by n-dimensional balls.

3 Computation of starting points

The clustering problem (1)-(4) can be reformulated in terms of nonsmooth, nonconvex optimization as follows (see (Bagirov et al. 2002, Bagirov et al. 2003)):

$$\text{minimize} \quad f(x) \tag{5}$$

subject to

$$x = (x^1, \ldots, x^k) \in \mathbb{R}^{n \times k}, \tag{6}$$

where

$$f(x^1, \ldots, x^k) = \frac{1}{m} \sum_{i=1}^{m} \min_{j=1,\ldots,k} \|x^j - a^i\|^2. \tag{7}$$

We call f a *cluster function.* If $k > 1$, the function f is nonconvex and nonsmooth. The number of variables in problem (1)-(4) is $(m+n) \times k$ whereas in problem (5)-(6) this number is only $n \times k$ and the number of variables does not depend on the number of instances. It should be noted that in many real-world data sets the number of instances m is substantially greater than the number of features n. On the other hand in the hard clustering problems the coefficients w_{ij} are integer, that is the problem (1)-(4) contains both integer and continuous variables. In the nonsmooth optimization formulation of the clustering problem variables are continuous only. All these circumstances can be considered as advantages of the nonsmooth optimization formulation (5)-(6) of the clustering problem.

Let us consider the problem of finding k-th cluster center assuming that the centers $x^1, \ldots, x^{k-1}$ for $k-1$ clusters are known. Then we introduce the following function:

$$\bar{f}^k(y) = \frac{1}{m} \sum_{i=1}^{m} \min \left\{ d^i_{k-1}, \|y - a^i\|^2 \right\} \tag{8}$$

where $y \in \mathbb{R}^n$ stands for k-th cluster center and

$$d^i_{k-1} = \min \left\{ \|x^1 - a^i\|^2, \ldots, \|x^{k-1} - a^i\|^2 \right\}.$$

The function $\bar{f}^k$ is called an *auxiliary cluster function.* It has only n variables.

Consider the set

$$\overline{D} = \left\{ y \in \mathbb{R}^n : \|y - a^i\|^2 \geq d^i_{k-1} \right\}.$$

$\bar{D}$ is the set where the distance between any its point y and any data point $a^i \in A$ is no less than the distance between this data point and its cluster center. We also consider the following set

$$D_0 = \mathbb{R}^n \setminus \overline{D} \equiv \{ y \in \mathbb{R}^n :$$

$$\exists I \subset \{1, \ldots, m\}, \ I \neq \emptyset : \|y - a^i\| < d^i_{k-1} \quad \forall i \in I \}.$$

The function $\bar{f}^k$ is a constant on the set $\overline{D}$ and its value in this set is

$$\bar{f}^k(y) = d_0 \equiv \sum_{i=1}^{m} d^i_{k-1}, \quad \forall y \in \overline{D}.$$

It is clear that $x^j \in \overline{D}$ for all $j = 1, \ldots, k-1$ and $a^i \in D_0$ for all $a^i \in A, \ \ a^i \neq x^j, \ j = 1, \ldots, k-1$. It is also clear that $\bar{f}^k(y) < d_0$ for all $y \in D_0$.

Any point $y \in D_0$ can be taken as a starting point for the k-th cluster center. The function $\bar{f}^k$ is nonconvex function with many local minima and one can assume that the global minimum of this function can be a good candidate to be the starting point for the k-th cluster center. However it is not always possible to find the global minimum of $\bar{f}^k$ in a reasonable time. Therefore we propose an algorithm for finding a local minimum of the function $\bar{f}^k$.

For any $y \in D_0$ we consider the following sets:

$$S_1(y) = \left\{ a^i \in A : \|y - a^i\|^2 = d^i_{k-1} \right\},$$

$$S_2(y) = \left\{ a^i \in A : \|y - a^i\|^2 < d^i_{k-1} \right\},$$

$$S_3(y) = \left\{ a^i \in A : \|y - a^i\|^2 > d^i_{k-1} \right\}.$$

The set $S_2(y) \neq \emptyset$ for any $y \in D_0$.

The the following algorithm is proposed to find a starting point for the k-th cluster center.

Algorithm 2 An algorithm for finding the starting point.

Step 1. For each $a^i \in D_0 \bigcap A$ compute the set $S_2(a^i)$, its center c^i and the value $\bar{f}^k_{a^i} = \bar{f}^k(c^i)$ of the function $\bar{f}^k$ at the point c^i.

Step 2. Compute

$$\bar{f}^k_{min} = \min_{a^i \in D_0 \bigcap A} \bar{f}^k_{a^i},$$

$$a^j = \arg\min{}_{a^i \in D_0 \bigcap A} \bar{f}^k_{a^i},$$

the corresponding center c^j and the set $S_2(c^j)$.

Step 3. Recompute the set $S_2(c^j)$ and its center until no more data points escape or return to this cluster.

Let $\bar{x}$ be a cluster center generated by Algorithm 2. Then the point $\bar{x}$ is a local minimum of the function $\bar{f}^k$.

4 An incremental clustering algorithm

In this section we describe an incremental algorithm for solving cluster analysis problems.

Algorithm 3 An incremental algorithm for clustering problems.

Step 1. (Initialization). Select a tolerance $\epsilon > 0$. Compute the center $x^{1*} \in \mathbb{R}^n$ of the set A. Let f^{1*} be the corresponding value of the objective function (7). Set $k = 1$.

Step 2. (Computation of the next cluster center). Let $x^{1*}, \ldots, x^{k*}$ be the cluster centers for k-partition problem. Apply Algorithm 2 to find a starting point $y^{k+1,0} \in \mathbb{R}^n$ for the $(k+1)$-st cluster center.

Step 3. (Refinement of all cluster centers). Take $x^{k+1,0} = (x^{1*}, \ldots, x^{k*}, y^{k+1,0})$ as a new starting point, apply k-means algorithm to solve $(k+1)$-partition problem. Let $x^{1*}, \ldots, x^{k+1,*}$ be a solution to this problem and $f^{k+1,*}$ be the corresponding value of the objective function (7).

Step 4. (Stopping criterion). If

$$\frac{f^{k*} - f^{k+1,*}}{f^{1*}} < \epsilon$$

then stop, otherwise set $k = k+1$ and go to Step 2.

It is clear that $f^{k*} \geq 0$ for all $k \geq 1$ and the sequence $\{f^{k*}\}$ is decreasing, that is,

$$f^{k+1,*} \leq f^{k,*} \quad \text{for all} \quad k \geq 1.$$

The latter implies that after $\bar{k} > 0$ iterations the stopping criterion in Step 4 will be satisfied. Thus Algorithm 3 computes as many clusters as the data set A contains with respect to the tolerance $\varepsilon > 0$.

The choice of the tolerance $\varepsilon > 0$ is crucial for Algorithm 3. Large values of ϵ can result in the appearance of large clusters whereas small values can produce small and artificial clusters.

5 Results of numerical experiments

To verify the effectiveness of the proposed algorithm and to compare it with similar algorithms a number of numerical experiments with six gene expression data sets have been carried out on a Pentium-4, 2.0 GHz, PC. We also use multi-start k-means (MSKM) and global k-means (GKM) algorithms for comparison. 100 randomly generated starting points are used in MSKM. In tables below MGKM stands for the modified global k-means algorithm. In tables we present the number of clusters (N), values f of the clustering function obtained by different algorithms and CPU time (t). We used the following gene expression data sets.

5.1 Data set 1

This data set is Boston Lung Cancer data set and was generated at the Dana Farber Cancer Institute. The data set consists of 12484 genes, 185 lung tumor samples and 17 normal lung samples. Of these, there were 138 lung adenocarcinoma, 6 small-cell lung cancer, 20 carcinoid lung cancer and 21 squamous cell. Expression profiles were generated using the Affymetrix GeneChip HG_U95Av2. This data set can be accessed from Cancer Genomics expression database at the Broad Institute of MIT and Harvard. Results for this data set are presented in Table 1.

Table 1: Results for Data set 1

N	MSKM		GKM		MGKM	
	$f \times 10^{10}$	t	$f \times 10^{10}$	t	$f \times 10^{10}$	t
2	8.441	542.81	8.441	59.31	8.441	102.47
5	6.644	1652.08	6.769	240.39	6.712	415.58
10	5.703	2714.59	6.094	545.19	5.696	962.94
15	5.467	4086.98	5.556	862.45	5.177	1543.30
20	4.900	5016.28	5.041	1199.98	4.812	2150.46

Results presented in Table 1 demonstrate that MSKM algorithm produces better results when the number of clusters $N \leq 10$. However MGKM outperforms two other algorithms as the number of clusters increases. GKM requires less CPU time however its solutions are not good. MGKM requires significantly less CPU time than MSKM.

5.2 Data set 2

This is the Novartis multi-tissue data set. The data set includes tissue samples of four cancer types with 26 breast,26 prostate, 28 lung, and 23 colon samples. There are 103 samples all together and 1000 genes. This data set is available at: http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi. Results for this data set are presented in Table 2.

One can see from Table 2 that algorithms repform similar when the number of clusters $N \leq 5$. However GKM requires significantly less CPU time. MGKM produces better solutions than two other algorithms as the number of clusters increases. Again MGKM requires less CPU time than MSKM.

Table 2: Results for Data set 2

N	MSKM		GKM		MGKM	
	$f \times 10^{10}$	t	$f \times 10^{10}$	t	$f \times 10^{10}$	t
2	9.212	0.81	9.212	0.19	9.212	0.30
5	5.024	3.30	5.032	0.61	5.032	1.03
10	3.424	6.70	3.408	1.36	3.351	2.88
15	2.849	10.13	2.897	2.16	2.812	5.98
20	2.470	11.42	2.556	3.00	2.422	10.23

5.3 Data set 3

This is a leukemia data set with 5000 genes and 38 samples including 11 acute myeloid leukemia (AML)and 27 acute lymphoblastic leukemia (ALL) samples. The original data set is retrievable from: http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi. Results are presented in Table 3. We calculate maximum 10 clusters because this data set contains only 38 samples.

Table 3: Results for Data set 3

N	MSKM		GKM		MGKM	
	$f \times 10^{10}$	t	$f \times 10^{10}$	t	$f \times 10^{10}$	t
2	7.880	3.06	8.137	0.58	7.880	0.67
5	5.537	8.17	5.837	2.02	5.729	2.64
10	4.104	10.47	4.399	4.59	4.271	8.19

Results from Table 3 show MSKM produces better solutions than two other algorithms, however it requires more computational time. MGKM produces better solutions than the GKM algorithm.

5.4 Data set 4

This data set includes 248 samples and 985 genes. Diagnostic bone narrow samples from pediatric acute leukemia patients corresponding to 6 prognostically important leukemia subtypes: 43 T-lineage ALL, 27 E2A-PBX1, 15 BCR-ABL, 79 TEL-AML1, 20 MLL rearrangements and 64 "hyperdiploid>50" chromosomes. The data set is available at: http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi. Computational results for this data set are presented in Table 4.

Table 4: Results for Data set 4

N	MSKM		GKM		MGKM	
	$f \times 10^{13}$	t	$f \times 10^{13}$	t	$f \times 10^{13}$	t
2	2.777	7.47	2.777	0.97	2.777	1.81
5	1.939	20.44	1.939	3.55	1.939	6.81
10	1.671	36.44	1.685	7.86	1.626	15.20
15	1.570	51.67	1.555	12.34	1.480	25.14
20	1.534	60.36	1.473	17.02	1.364	36.09

For data set 4 all three algorithms give the same solutions when the number of clusters $N \leq 5$. However, for larger number of clusters MGKM outperforms other two algorithms. GKM requires the least CPU time and MGKM requires less CPU time than MSKM.

5.5 Data set 5

This is a lung cancer data set which includes 2000 genes and 139 adenocarcinomas, 21 squamous cell carcinomas, 20 carcinoids and 17 normal lung samples. This data set is available at: http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi. Results are given in Table 5.

Table 5: Results for Data set 5

N	MSKM		GKM		MGKM	
	$f \times 10^{10}$	t	$f \times 10^{10}$	t	$f \times 10^{10}$	t
2	1.588	5.28	1.589	0.70	1.589	1.23
5	1.068	24.30	1.067	2.33	1.067	4.47
10	0.870	39.94	0.880	5.27	0.862	10.05
15	0.860	50.67	0.819	8.23	0.781	15.61
20	0.824	53.45	0.766	11.23	0.726	22.47

Results presented in Table 5 demonstrate that algorithms produce almost the same solutions when the number of clusters $N \leq 5$. The algorithm MGKM significantly outperforms other algorithms as the number of clusters increases. GKM requires the least CPU time and MGKM requires less CPU time than the algorithm MSKM.

5.6 Data set 6

This data set has 90 samples and 1277 genes. It contains 13 distinct tissue types: 5 breast cancer, 9 prostate, 7 lung, 11 colon, 6 germinal center cells, 7 bladder, 6 uterus, 5 peripheral blood monocytes, 12 kidney, 10 pancreas, 4 ovary, 5 whole brain and 3 cerebellum. This data set is available at: http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi. Computational results for this data set are presented in Table 6.

Table 6: Results Data set 6

N	MSKM		GKM		MGKM	
	$f \times 10^{11}$	t	$f \times 10^{11}$	t	$f \times 10^{11}$	t
2	1.554	2.16	1.589	0.20	1.582	0.36
5	1.040	7.06	1.064	0.69	1.065	1.23
10	0.655	14.28	0.651	1.52	0.633	2.69
15	0.526	23.58	0.461	2.44	0.453	4.86
20	0.476	29.78	0.352	3.38	0.349	8.27

Results from Table 6 demonstrate that for small number clusters MSKM works better than other algorithms, however GKM and MGKM produce better solutions as the number of clusters increases. MGKM is best for larger number clusters. MSKM is computationally more expensive and GKM use the least CPU time.

5.7 Content of clusters

In this subsection we demonstrate the content of clusters produced by different algorithms and we use the notion of cluster purity to compare clusters. The notion of cluster purity is defined as follows:

$$P(A^i) = 100 \frac{1}{n_{A^i}} \max_{j=1,\dots,l} n^j_{A^i},$$

where $n_{A^i} = |A^i|$ is the cardinality of the cluster A^i, $n^j_{A^i}$ is the number of instances in the cluster A^i that belong to the true class j and l is the number of true classes. Then the total purity $P(A)$ for the data set A can be calculated as:

$$P(A) = \frac{n_{A^i} P(A^i)}{m}.$$

We used the data set 6 and calculated 30 clusters. Results are as follows.

- MSKM algorithm produced 13 empty, 6 mixed and 11 pure clusters with total purity $P(A) = 64.44$;
- GKM algorithm produced 27 pure and 3 mixed clusters with the total purity $P(A) = 83.33$. In three mixed clusters the results were as follows:
  - Cluster 1 - 17 tumors: breast(1), lung(2), colon(2), germinal center cells (1), bladder(1), uterus(2), kidney(3), pancreas(5);
  - Cluster 2 - 4 tumors: bladder(1), uterus(3);
  - Cluster 3 - 5 tumors: whole brain(2), cerebellum(3).
- MGKM algorithm produced 27 pure and 3 mixed clusters with the total purity $P(A) = 85.56$. In three mixed clusters the results were as follows:
  - Cluster 1 - 14 tumors: breast(1), lung(2), colon(1), bladder(2), kidney(3), pancreas(5);
  - Cluster 2 - 3 tumors: colon(1), germinal center cells (1), bladder(1).
  - Cluster 3 - 5 tumors: bladder(1), uterus(3), whole brain(1);

One can see that MGKM algorithm produces better clusters than two other algorithms.

6 Conclusions

In this paper we have developed the new version of the global k-means algorithm, the modified global k-means algorithm. This algorithm computes clusters incrementally and to compute k-partition of a data set it uses $k-1$ cluster centers from the previous iteration. An important step in this algorithm is the computation of a starting point for the k-th cluster center. This starting point is computed by minimizing so-called auxiliary cluster function. The proposed algorithm computes as many clusters as a data set contains with respect to a given tolerance.

We have presented the results of numerical experiments on 6 gene expression data sets. These results clearly demonstrate that the modified global k-means algorithm proposed in this paper is efficient for solving clustering problems in gene expression data sets. It outperforms both the multi-start and global k-means algorithms as the number of clusters increases. However the proposed algorithm requires more computational efforts than the global k-means algorithm.

Acknowledgement

This research was supported by the Australian research Council.

References

Al-Sultan, K.S. (1995), A tabu search approach to the clustering problem, *Pattern Recognition*, **28(9)**, 1443-1451.

Bagirov, A.M., Rubinov, A.M. & Yearwood, J. (2002), A global optimisation approach to classification, *Optimization and Engineering,* **3(2)**, 129-155.

Bagirov, A.M., Rubinov, A.M, Soukhoroukova, N.V. & Yearwood, J. (2003), Supervised and unsupervised data classification via nonsmooth and global optimisation, *TOP: Spanish Operations Research Journal,* **11(1)**, 1-93.

Bagirov, A.M. & Yearwood, J. (2006), A new nonsmooth optimization algorithm for minimum sum-of-squares clustering problems, *European Journal of Operational Research,* **170(2)**, 578-596.

Bagirov, A.M., Ferguson, B., Ivkovic, S., Saunders, G. & Yearwood, J. (2003), New algorithms for multi-class cancer diagnosis using tumor gene expression signatures, *Bioinformatics,* **19(14)**, 1800-1807.

Bock, H.H. (1998), Clustering and neural networks, In: Rizzi, A., Vichi, M. & Bock, H.H. (eds), *Advances in Data Science and Classification*, Springer-Verlag, Berlin, pp. 265-277.

Brown, D.E. & Entail, C.L. (2001), A practical application of simulated annealing to the clustering problem, *Pattern Recognition*, **25(4)**, 401-412.

de Merle, O., Hansen, P., Jaumard, B. & Mladenovic, N. (2001), An interior point method for minimum sum-of-squares clustering, *SIAM J. on Scientific Computing,* **21,** 1485-1505.

Diehr, G. (1985), Evaluation of a branch and bound algorithm for clustering, *SIAM J. Scientific and Statistical Computing*, **6,** 268-284.

Dubes, R. & Jain, A.K. (1976), Clustering techniques: the user's dilemma, *Pattern Recognition*, **8,** 247-260.

Hanjoul, P. & Peeters, D. (1985), A comparison of two dual-based procedures for solving the p-median problem, *European Journal of Operational Research,* **20,** 387-396.

Hansen, P. & Jaumard, B. (1997), Cluster analysis and mathematical programming, *Mathematical Programming,* **79(1-3),** 191-215.

Hansen, P. & Mladenovic, N. (2001a), J-means: a new heuristic for minimum sum-of-squares clustering, *Pattern Recognition*, **4,** 405-413.

Hansen, P. & Mladenovic, N. (2001b), Variable neighborhood decomposition search, *Journal of Heuristic,* **7,** 335-350.

Hansen, P., Ngai, E., Cheung, B.K. & Mladenovic, N. (2001b), Analysis of global k-means, an incremental heuristic for minimum sum-of-squares clustering, submitted.

Houkins, D.M. , Muller, M.W. & ten Krooden, J.A., (2001b), Cluster analysis, In: *Topics in Applied Multivariate Analysis*, Cambridge University press, Cambridge.

Jain, A.K. , Murty, M.N. & Flynn, P.J. (1999), Data clustering: a review, *ACM Computing Surveys,* **31(3),** 264-323.

Jensen, R.E. (1969), A dynamic programming algorithm for cluster analysis, *Operations Research,* **17,** 1034-1057.

Koontz, W.L.G., Narendra, P.M. & Fukunaga, K. (1975), A branch and bound clustering algorithm, *IEEE Transactions on Computers*, **24,** 908-915.

Likas, A., Vlassis, M. & Verbeek, J. (2003), The global k-means clustering algorithm, *Pattern Recognition*, **36,** 451-461.

Medvedovic, M. & Sivaganesan, S. (2002), Bayesian infinite mixture model based clustering gene expression profiles, *Bioinformatics*, **18,** 1194-1206.

Selim, S.Z. & Al-Sultan, K.S. (1991), A simulated annealing algorithm for the clustering, *Pattern Recognition*, **24(10)**, 1003-1008.

Spath, H. (1991), *Cluster Analysis Algorithms*, Ellis Horwood Limited, Chichester.

Sun, L.X., Xie, Y.L., Song, X.H., Wang, J.H. & Yu, R.Q. (1994), Cluster analysis by simulated annealing, *Computers and Chemistry*, **18,** 103-108.

Yeung, K.Y., Fraley, C., Murua, A., Raftery, A.E. & Ruzzo, W.L. (2001), Model-based clustering and data transformations for gene expression data, *Bioinformatics,* **17,** 977-987.

Yeung, K.Y. , Medvedovic, M. & Bumgarner, R.E., (2003), Clustering gene expression data with repeated measurements, *Genome Biol.,* **4,** R34.

Clustering Replicated Microarray Data via Mixtures of Random Effects Models for Various Covariance Structures

S.K. Ng[1], **G.J. McLachlan**[1,2], **R.W. Bean**[1,2], **and S.-W. Ng**[3]

[1] Department of Mathematics, University of Queensland, Brisbane, QLD 4072, Australia
Email: {skn, gjm, rbean}@maths.uq.edu.au
[2] Institute for Molecular Bioscience, University of Queensland, Brisbane, QLD 4072, Australia
[3] Laboratory of Gynecologic Oncology, Department of Obstetrics, Gynecology and Reproductive Biology,
Brigham and Women's Hospital, Boston, MA 02115, USA
Email: sng@rics.bwh.harvard.edu

Abstract

A unified approach of mixed-effects model has been recently proposed for clustering correlated genes from different kinds of microarray experiments. With the so-called **EM**-based **MIX**ture analysis **WI**th **R**andom **E**ffects (EMMIX-WIRE) model, both the gene-specific and tissue-specific random effects are taken into account in the (mixture) modelling of microarray data. In this paper, we focus on the applications of the EMMIX-WIRE model to the cluster analysis of microarray data with repeated measurements. In particular, we investigate various forms of covariance structure commonly applicable for replicated microarray data and compare their impact on the final clustering results, using a real data set of microRNA profile and a published yeast galactose data set with known Gene Ontology (GO) listings.

Keywords: EMMIX-WIRE model, Random effects models, Covariance structures, Replicated microarray data.

1 Introduction

The advent of high-throughput technologies has revolutionized molecular biology, and indeed is setting the stage for the rapid evolution of the way disease is diagnosed, classified, and treated. The complexity of tumours makes it likely that a diagnostic test will be based on marker profiles rather than individual markers. However, the identification of relevant subsets of the markers has its challenges, because microarray experiments are now being carried out with replication for capturing either biological or technical variability in expression levels to improve the quality of inferences made from experimental studies (Lee, Kuo, Whitmore & Sklar 2000, Pavlidis, Li & Noble 2003). Replicated measurements of gene expression for a microarray experiment are often correlated and tend to be more alike in characteristics than measurements for the microarray experiments as a whole. At the same time, gene expression levels from the same experiment are correlated (McLachlan, Do & Ambroise 2004). It means that clustering methods which assume independently distributed gene profiles should produce less reliable results than those that exploit or allow for correlation between the gene profiles. Indeed, ignoring the dependence between the gene profiles and the covariance structure of replicated microarray data can result in important sources of variability in the experiments being overlooked in the analysis, with the consequent possibility of misleading inferences being made (McLachlan et al. 2004, Ng, McLachlan, Wang, Ben-Tovim & Ng 2006).

Mixed-effects models have been used in the model-based cluster analysis of gene expression data from time-course experiments and experiments with repeated measurements (Luan & Li 2003, Celeux, Martin & Lavergne 2005). However, with these mixed-effects models, only the correlation between replicated measurements for a gene from each microarray experiment is considered (by modelling via gene-specific random effects). Thus, these models require the independence assumption for the genes which, however, will not hold in practice for all pairs of genes (McLachlan et al. 2004, Klebanov, Jordan & Yakovlev 2006) because of the correlation between gene expression levels from the same microarray experiment (tissue-specific effects). Recently, a unified approach of mixed-effects model has been proposed for clustering correlated genes from different kinds of microarray experiments, where both the gene-specific and tissue-specific random effects (Ng et al. 2006) are taken into account in the (mixture) modelling of microarray data. With this so-called **EM**-based **MIX**ture analysis **WI**th **R**andom **E**ffects (EMMIX-WIRE) approach, the unknown model parameters can be obtained by maximum likelihood (ML) via the Expectation-Maximization (EM) algorithm of Dempster *et al.* (1977). see also Ng, Krishnan & McLachlan (2004).

In this paper, we focus on applications of the EMMIX-WIRE procedure to the cluster analysis of microarray data with repeated measurements. In particular, we investigate various forms of covariance structure commonly used for replicated microarray data and compare their impacts on the final clustering results. The rest of the paper is organized as follows: Section 2 introduces the EMMIX-WIRE model for clustering microarray data with repeated measurements and outlines the ML estimation via the EM algorithm. In Section 3, various forms of covariance structure for replicated microarray data are considered and discussed. The impact of various covariance structures on the cluster analysis is studied in Section 4, using a real data set of microRNA profile and a published yeast galactose data set with known Gene Ontology (GO) listings (Ashburner et al. 2000). Section 5 ends the paper with some discussion.

This work was supported by grant from the Australian Research Council.

2 EMMIX-WIRE model for cluster analysis

The EMMIX-WIRE procedure of Ng *et al.* (2006) formulates a linear mixed-effects model (LMM) (McCulloch & Searle 2001) for the mixture components in which covariate information can be incorporated into the clustering process. For t biological samples (not all necessarily independent) with r replicate hybridizations for each, we let $\boldsymbol{y}_j = (\boldsymbol{y}_{1j}^T, \ldots, \boldsymbol{y}_{tj}^T)^T$ contain the expression levels for the jth gene, where

$$\boldsymbol{y}_{lj} = (y_{l1j}, \ldots, y_{lrj})^T \qquad (l = 1, \ldots, t)$$

contains the r technical replicates for the lth biological sample on the jth gene. The superscript T above denotes vector transpose. It is assumed that the (logged) expression levels have been preprocessed with adjustment for array effects. The microarray data can be therefore represented by an $n \times m$ matrix, where $m = t \times r$ is the dimension of the gene-expression profiles. With the EMMIX-WIRE procedure, the observed m-dimensional vectors $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$ are assumed to have come from a mixture of a finite number, say g, of components in some unknown proportions $\pi_1, \ldots, \pi_g$, which sum to one. Conditional on its membership of the ith component of the mixture, the vector $\boldsymbol{y}_j$ for the jth gene ($j = 1, \ldots, n$) follows the model

$$\boldsymbol{y}_j = \boldsymbol{X}\boldsymbol{\beta}_i + \boldsymbol{U}\boldsymbol{b}_{ij} + \boldsymbol{V}\boldsymbol{c}_i + \boldsymbol{\epsilon}_{ij}, \tag{1}$$

where the elements of $\boldsymbol{\beta}_i$ (an t-dimensional vector) are fixed effects (unknown constants) modelling the conditional mean of $\boldsymbol{y}_j$ in the ith component ($i = 1, \ldots, g$). In (1), $\boldsymbol{b}_{ij}$ (an q_b-dimensional vector) and $\boldsymbol{c}_i$ (an q_c-dimensional vector) represent the unobservable gene- and tissue-specific random effects, respectively. These random effects represent the variation due to the heterogeneity of genes and samples (corresponding to $\boldsymbol{b}_i = (\boldsymbol{b}_{i1}^T, \ldots, \boldsymbol{b}_{in}^T)^T$ and $\boldsymbol{c}_i$, respectively). The random effects $\boldsymbol{b}_i$ and $\boldsymbol{c}_i$, and the measurement error vector $(\boldsymbol{\epsilon}_{i1}^T, \ldots, \boldsymbol{\epsilon}_{in}^T)^T$ are assumed to be mutually independent, where $\boldsymbol{X}$, $\boldsymbol{U}$, and $\boldsymbol{V}$ are known design matrices of the corresponding fixed or random effects, respectively.

With the LMM, the distributions of $\boldsymbol{b}_{ij}$ and $\boldsymbol{c}_i$ are taken, respectively, to be multivariate normal $N_{q_b}(\boldsymbol{0}, \theta_{bi}\boldsymbol{I}_{q_b})$ and $N_{q_c}(\boldsymbol{0}, \theta_{ci}\boldsymbol{I}_{q_c})$, where $\boldsymbol{I}_{q_b}$ and $\boldsymbol{I}_{q_c}$ are identity matrices with dimensions being specified by the subscripts. The measurement error vector $\boldsymbol{\epsilon}_{ij}$ is also taken to be multivariate normal $N_m(\boldsymbol{0}, \boldsymbol{A}_i)$, where $\boldsymbol{A}_i = \mathrm{diag}(\boldsymbol{W}\boldsymbol{\phi}_i)$ is a diagonal matrix constructed from the vector $(\boldsymbol{W}\boldsymbol{\phi}_i)$ with $\boldsymbol{\phi}_i = (\sigma_{i1}^2, \ldots, \sigma_{iq_e}^2)^T$ and $\boldsymbol{W}$ a known $m \times q_e$ zero-one design matrix. That is, we allow the ith component-variance to be different among the m hybridizations.

We let $\boldsymbol{\Psi} = (\boldsymbol{\psi}_1^T, \ldots, \boldsymbol{\psi}_g^T, \pi_1, \ldots, \pi_{g-1})^T$ be the vector of all the unknown parameters, where $\boldsymbol{\psi}_i$ is the vector containing the unknown parameters $\boldsymbol{\beta}_i$, θ_{bi}, θ_{ci}, and $\boldsymbol{\phi}_i$ of the ith component density ($i = 1, \ldots, g$). The estimation of $\boldsymbol{\Psi}$ can be obtained by the ML approach via the EM algorithm, proceeding conditionally on the tissue-specific random effects $\boldsymbol{c}_i$ as formulated in Ng *et al.* (2006). The E- and M-steps can be implemented in closed form. In particular, an approximation to the E-step by carrying out time-consuming Monte Carlo methods is not required. A probabilistic or an outright clustering of the genes into g components can be obtained, based on the estimated posterior probabilities of component membership given the profile vectors and the estimated tissue-specific random effects $\hat{\boldsymbol{c}}_i$ for $i = 1, \ldots, g$; see Ng *et al.* (2006).

3 Covariance structures for replicated experiments

Let $\boldsymbol{Y}^i$ denote a random vector of size $n_i m$ consisting of all the observations $\boldsymbol{y}_j$ that arise from the ith component, where n_i is the number of genes belonging to the ith component. It is assumed that all $\boldsymbol{y}_j$ in the ith component are independent given $\boldsymbol{c}_i$. The conditional distribution of $\boldsymbol{Y}^i \mid \boldsymbol{c}_i$ is then given by $N_{n_i m}(\boldsymbol{\Lambda}_i\boldsymbol{\beta}_i, \boldsymbol{\Sigma}_i)$, where $\boldsymbol{\Lambda}_i = (\boldsymbol{1}_{n_i} \otimes \boldsymbol{X})$. Here, $\boldsymbol{1}_{n_i}$ is an n_i-dimensional vector of ones, the sign $\otimes$ is the Kronecker product of two matrices, and

$$\boldsymbol{\Sigma}_i = \boldsymbol{I}_{n_i} \otimes (\boldsymbol{A}_i + \theta_{bi}\boldsymbol{U}\boldsymbol{U}^T).$$

Hence, the unconditional distribution of $\boldsymbol{Y}^i$ is given by $N_{n_i m}(\boldsymbol{\Lambda}_i\boldsymbol{\beta}_i, \boldsymbol{\Sigma}_i + \boldsymbol{J}_{n_i} \otimes \boldsymbol{D}_i)$, where $\boldsymbol{J}_{n_i}$ is an $n_i \times n_i$ matrix of ones and

$$\boldsymbol{D}_i = \theta_{ci}\boldsymbol{V}\boldsymbol{V}^T. \tag{2}$$

The presence of the term (2) in the covariance matrix of $\boldsymbol{Y}^i$ induces the correlation between genes that belong to the same cluster.

For the specification of gene-specific random effects $\boldsymbol{b}_{ij}$, we consider two typical models applicable for replicated microarray data. The first model takes $\boldsymbol{U} = \boldsymbol{X}$ and $q_b = t$ such that $\boldsymbol{b}_{ij} = (b_{i1j}, \ldots, b_{itj})^T$. That is, it is assumed that a gene-specific random effect, b_{ilj}, is shared among the repeated measurements of expression on the jth gene in the lth biological sample ($l = 1, \ldots, t$). The replicated measurements are therefore correlated. The second model simplifies the first one by taking $\boldsymbol{U} = \boldsymbol{1}_m$ and $q_b = 1$. That is, it is assumed that a gene-specific random effect, b_{ij}, is shared among the measurements on the jth gene from all the $m = t \times r$ hybridizations.

For the specification of tissue-specific random effects $\boldsymbol{c}_i$, we consider three typical models applicable for replicated microarray data. The first model takes $\boldsymbol{V} = \boldsymbol{I}_m$ and $q_c = m = t \times r$ such that $\boldsymbol{c}_i = (c_{i11}, \ldots, c_{ir1}, \ldots, c_{i1t}, \ldots, c_{irt})^T$. That is, it is assumed that a tissue-specific random effect, c_{ikl}, is shared among gene expressions from the kth replicate of the lth biological sample ($k = 1, \ldots, r$; $l = 1, \ldots, t$). It means that genes within the same cluster are correlated. In some microarray experiments, the t biological samples, however, are not all independent. For example, they could correspond to samples from p patients with $t_1 + t_2 + \ldots + t_p = t$. The value t_s corresponds to the number of biological samples from the sth patient ($s = 1, \ldots, p$). For example, the t_s biological samples for the sth patient might correspond to samples taken at t_s different time points or in t_s different conditions. A second model can be adopted to incorporate such a data hierarchy by taking

$$\boldsymbol{V} = \boldsymbol{V}^* = \begin{pmatrix} \boldsymbol{1}_{t_1 r} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{1}_{t_2 r} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1}_{t_p r} \end{pmatrix},$$

and $q_c = p$. It means that a patient-specific random effect, c_{is}, is shared among gene expression levels for the technical and biological replicates for the sth patient ($s = 1, \ldots, p$). It thus induces a correlation between the expression levels of different genes on the same patient provided the genes belong to the same cluster. The third model simplifies the above two models by taking $\boldsymbol{V} = \boldsymbol{0}$. That is, it is assumed that there are no tissue-specific random effects and genes are not correlated (an independence model).

By considering the combinations of the above random-effects models, we have six forms of covariance structures:

Model 1: Taking $\boldsymbol{U} = \boldsymbol{X}$, $q_b = t$, $\boldsymbol{V} = \boldsymbol{I}_m$, $q_c = m$, $\boldsymbol{W} = \boldsymbol{X}$, and $q_e = t$, the covariance matrix for the unconditional distribution of $\boldsymbol{Y}^i$ is given by

$$\boldsymbol{I}_{n_i} \otimes (\text{diag}(\boldsymbol{X}\boldsymbol{\phi}_i) + \theta_{bi}\boldsymbol{X}\boldsymbol{X}^T) + \boldsymbol{J}_{n_i} \otimes \theta_{ci}\boldsymbol{I}_m\boldsymbol{I}_m^T,$$

where $\boldsymbol{\phi}_i = (\sigma_{i1}^2, \ldots, \sigma_{it}^2)^T$.

Model 2: Taking $\boldsymbol{U} = \boldsymbol{X}$, $q_b = t$, $\boldsymbol{V} = \boldsymbol{V}^*$, $q_c = p$, $\boldsymbol{W} = \boldsymbol{X}$, and $q_e = t$, the covariance matrix for $\boldsymbol{Y}^i$ is given by

$$\boldsymbol{I}_{n_i} \otimes (\text{diag}(\boldsymbol{X}\boldsymbol{\phi}_i) + \theta_{bi}\boldsymbol{X}\boldsymbol{X}^T) + \boldsymbol{J}_{n_i} \otimes \theta_{ci}\boldsymbol{V}^*\boldsymbol{V}^{*T},$$

where $\boldsymbol{\phi}_i = (\sigma_{i1}^2, \ldots, \sigma_{it}^2)^T$.

Model 3: Taking $\boldsymbol{U} = \boldsymbol{X}$, $q_b = t$, $\boldsymbol{V} = \boldsymbol{0}$, $\boldsymbol{W} = \boldsymbol{X}$, and $q_e = t$, the covariance matrix for $\boldsymbol{Y}^i$ is given by

$$\boldsymbol{I}_{n_i} \otimes (\text{diag}(\boldsymbol{X}\boldsymbol{\phi}_i) + \theta_{bi}\boldsymbol{X}\boldsymbol{X}^T),$$

where $\boldsymbol{\phi}_i = (\sigma_{i1}^2, \ldots, \sigma_{it}^2)^T$.

Model 4: Taking $\boldsymbol{U} = \boldsymbol{1}_m$ and $q_b = 1$, $\boldsymbol{V} = \boldsymbol{I}_m$, $q_c = m$, $\boldsymbol{W} = \boldsymbol{1}_m$, and $q_e = 1$, the covariance matrix for $\boldsymbol{Y}^i$ is given by

$$\boldsymbol{I}_{n_i} \otimes (\text{diag}(\boldsymbol{1}_m\phi_i) + \theta_{bi}\boldsymbol{1}_m\boldsymbol{1}_m^T) + \boldsymbol{J}_{n_i} \otimes \theta_{ci}\boldsymbol{I}_m\boldsymbol{I}_m^T,$$

where $\phi_i = \sigma_i^2$.

Model 5: Taking $\boldsymbol{U} = \boldsymbol{1}_m$ and $q_b = 1$, $\boldsymbol{V} = \boldsymbol{V}^*$, $q_c = p$, $\boldsymbol{W} = \boldsymbol{1}_m$, and $q_e = 1$, the covariance matrix for $\boldsymbol{Y}^i$ is given by

$$\boldsymbol{I}_{n_i} \otimes (\text{diag}(\boldsymbol{1}_m\phi_i) + \theta_{bi}\boldsymbol{1}_m\boldsymbol{1}_m^T) + \boldsymbol{J}_{n_i} \otimes \theta_{ci}\boldsymbol{V}^*\boldsymbol{V}^{*T},$$

where $\phi_i = \sigma_i^2$.

Model 6: Taking $\boldsymbol{U} = \boldsymbol{1}_m$, $q_b = 1$, $\boldsymbol{V} = \boldsymbol{0}$, $\boldsymbol{W} = \boldsymbol{1}_m$, and $q_e = 1$, the covariance matrix for $\boldsymbol{Y}^i$ is given by

$$\boldsymbol{I}_{n_i} \otimes (\text{diag}(\boldsymbol{1}_m\phi_i) + \theta_{bi}\boldsymbol{1}_m\boldsymbol{1}_m^T),$$

where $\phi_i = \sigma_i^2$.

To examine the (biological) meaning of Equation (1) for the various models above, we consider Model 1. Under this model, it is assumed that the expression level of the jth gene, conditional of its membership of the ith component of the mixture (ith cluster), is given for the kth replicate in the lth experiment by

$$y_{jkl} = \beta_{il} + b_{ilj} + c_{ikl} + \epsilon_{ijkl}$$

($i = 1, \ldots, g; j = 1, \ldots, n; k = 1, \ldots, r; l = 1, \ldots, t$). That is, the expression level y_{jkl} is equal to the mean expression level at the lth experiment for the ith component (β_{il}) plus a gene-specific random effect b_{ilj}, a tissue-specific random effect c_{ikl}, and an experimental random error ϵ_{ijkl}. The vector of dimension $q_b = t$, $(b_{i1j}, \ldots, b_{itj})^T$, represents the variation between the gene expression profiles and their component-means for the t microarray experiments. The vector of dimension $q_c = m$, $(c_{i11}, \ldots, c_{ir1}, \ldots, c_{i1t}, \ldots, c_{irt})^T$, represents the variation between expression signature and the component-mean signature for the $m = t \times r$ hybridizations.

It can be seen from the covariance matrix for $\boldsymbol{Y}^i$ that Models 3 and 6 are independence models, where there are no tissue-specific random effects being assumed ($\boldsymbol{V} = \boldsymbol{0}$). It means that expression levels for the same microarray experiment are independent.

4 Comparative studies

The impact of various covariance structures on the cluster analysis is compared using a real data set of microRNA profile and a published yeast galactose data set with known GO listings.

MicroRNAs are a family of small (~22 nucleotides) noncoding RNA molecules that are evolutionary conserved and are expressed in a tissue-specific and developmental stage-specific manner (Bartel 2004). They are important regulators of various aspects of developmental control in both plants and animals through sequence-specific interactions with target mRNAs. Recent studies have shown that microRNA expression profiles are more accurate than global mRNA profiles in classifying the histologic origins and differentiation of human tumours (Lu et al. 2005) and highlighted the potential of microRNA profiling in cancer diagnosis and classification (He et al. 2005). The data set consists of three ($r = 3$) replicate hybridizations for each microRNA microarray experiment of $t = 12$ samples. However, there is a large amount of missing data. We therefore work with a subset of $n = 160$ microRNAs that have about 13% of the data missing. All the missing expressions were imputed using the support vector regression (SVR) imputation and orthogonal coding scheme (Wang, Li, Jiang & Feng 2006). We are interested primarily in which microRNAs are put together in the same cluster for plausible choices of the number of components g in the mixture model. A guide to plausible values of g can be obtained using the Bayesian information criterion (BIC) of Schwarz (1978). This criterion, which is based on a penalized form of the log likelihood, has growing support in the literature for selecting the value of g in the context of mixture model-based clustering of microarray data (Luan & Li 2003, Yeung, Fraley, Murua, Raftery & Ruzzo 2001); see also the discussion in Ng *et al.* (2006).

The covariance structures in Models 1, 3, 4, and 6 presented in Section 3 are considered now. Models 2 and 5 were not considered as there was no information available on the experiments to suggest that they would be applicable. As an illustration for Model 1, we take $m = t \times r = 36$ and $\boldsymbol{X} = \boldsymbol{1}_3 \otimes \boldsymbol{I}_{12}$ (a 36 × 12 matrix). The design matrices $\boldsymbol{U}$, $\boldsymbol{V}$, and $\boldsymbol{W}$ are taken to be equal to $\boldsymbol{X}$, $\boldsymbol{I}_{36}$, and $\boldsymbol{X}$, respectively. We fit this model for various values of the number of components g. Model selection via BIC indicated that there are five clusters.

Based on the setting of $g = 5$, we then fit the mixed-effects models with various covariance structures. The clusters so formed are then compared to that obtained from Model 1 above. The adjusted Rand index (Hubert & Arabie 1985) is adopted to assess the degree of agreement between two clustering partitions. A larger adjusted Rand index indicates a higher level of agreement. Identical clustering partitions will have the adjusted Rand index of one. In Table 1, the adjusted Rand indices for various covariance structures considered are presented. It can be seen that various covariance structures did result in different clustering of microRNAs.

To illustrate further the relative impact of the

Table 1: Adjusted Rand indices with reference to the clustering obtained from Model 1 (MicroRNA data)

Covariance structure	Adjusted Rand index
Model 1	1.0
Model 3	0.723
Model 4	0.298
Model 6	0.298

Table 2: Adjusted rand indices with reference to the known GO listings (Yeast galactose data)

Covariance structure	Adjusted rand index
Model 1	0.978
Model 3	0.811
Model 4	0.906
Model 6	0.910

adopted covariance structure on the cluster analysis, we work on a published yeast data set with known GO listings (Ideker et al. 2001, Yeung, Medvedovic & Bumgarner 2003). With this yeast galactose data, there are four ($r = 4$) replicate hybridizations for each cDNA array experiment. There are $n = 205$ genes and $t = 20$ microarray experiments. The expression patterns of these 205 genes reflect four functional categories in the GO listings (Yeung et al. 2003). We first applied Model 1 given in Section 3 to cluster the genes into $g = 4$ groups. The clusters so formed are then compared to the four categories in the GO listings. The adjusted Rand index was found to be 0.978, which is the best match (the largest index) compared with several model-based and hierarchical clustering algorithms considered in Yeung *et al.* (2003). The adjusted Rand indices for mixed-effects models with various covariance structures are given in Table 2. Again, it can be seen that different clustering results are obtained from the various covariance structures considered.

5 Discussion

We have investigated various covariance structures in EMMIX-WIRE model applicable for clustering replicated microarray data. The specification of covariance structures needs careful consideration. The choice should be justified by the data hierarchy so formed due to the design of microarray experiments. With repeated measures data, replicated measurements of size r from t microarray experiments on each gene are obtained. It is therefore anticipated that random effects are shared among expression levels to represent the variation due to the heterogeneity of genes and samples (corresponding to $\boldsymbol{b}_i$ and $\boldsymbol{c}_i$, respectively), as discussed in Section 3. It is interesting to note that combinations of random effects may be considered in mixed-effects modelling. For example, an alternative model for the specification of gene-specific random effects $\boldsymbol{b}_{ij}$ may be adopted by combining the two models $\boldsymbol{U} = \boldsymbol{X}$ and $\boldsymbol{U} = \mathbf{1}_m$ together (that is, a random effect accounting for correlation among replicated measurements plus another accounting for correlation among all hybridizations). However, it was demonstrated in Celeux *et al.* (2005) that this model provided quite similar results to that of the first model with $\boldsymbol{U} = \boldsymbol{X}$. This result indicates that combinations of random effects are usually not required.

The impact of various covariance structures on the clustering results are compared in Section 4. It can be seen that Model 1 outperforms others for the cluster analysis of the yeast galactose data. With Model 1, it is assumed that a gene-specific random effect, b_{ilj}, is shared among the repeated measurements on the jth gene from the lth microarray experiment ($l = 1, \ldots, t$). A tissue-specific random effect, c_{ikl}, is also assumed to be shared among gene expressions from the kth replicate for the lth experiment ($k = 1, \ldots, r; l = 1, \ldots, t$). It means that replicated measurements are correlated and genes within the same cluster are also correlated. This correlation structure is justified by the data hierarchy so formed in typical replicated microarray experiments. On the other hand, the simplified model for the specification of $\boldsymbol{b}_{ij}$ with $\boldsymbol{U} = \mathbf{1}_m$ can be regarded as unrealistic in many situations of replicated microarray experiments (Celeux et al. 2005), as indicated in Table 1 for the microRNA data.

References

Ashburner, M., Ball, C.A., Blake, J.A. et al. (2000), ‘Gene Ontology: tool for the unification of biology’, *Nat. Genet.* **25**, 25–29.

Bartel, D.P. (2004), ‘MicroRNAs: genomics, biogenesis, mechanism, and function’, *Cell* **116**, 281–297.

Celeux, G., Martin, O. & Lavergne, C. (2005), ‘Mixture of linear mixed models for clustering gene expression profiles from repeated microarray experiments’, *Statistical Modelling* **5**, 243–267.

Dempster, A.P., Laird, N.M. & Rubin, D.B. (1977), ‘Maximum likelihood from incomplete data via the EM algorithm’, *J. Roy. Stat. Soc. Ser. B* **39**, 1–38.

He, L., Thomson, J.M., Hemann, M.T., et al. (2005), ‘A microRNA polycistron as a potential human oncogene’, *Nature* **435**, 828–833.

Hubert, L. & Arabie, P. (1985), ‘Comparing partitions’, *J. Classif.* **2**, 193–218.

Ideker, T., Thorsson, V., Ranish, J.A., et al. (2001), ‘Integrated genomic and proteomic analyses of a systemically perturbed metabolic network’, *Science* **292**, 929–934.

Klebanov, L., Jordan, C. & Yakovlev, A. (2006), ‘A new type of stochastic dependence revealed in gene expression data’, *Stat. Appl. Genetics Mol. Biol.* **5**, No. 1, Article 7.

Lee, M.L.T., Kuo, F.C., Whitmore, G.A. & Sklar, J. (2000), ‘Importance of replication in microarray gene expression studies: statistical methods and evidence from repetitive cDNA hybridizations’, *Proc. Natl. Acad. Sci. USA* **97**, 9834–9838.

Lu, Y., Getz, G., Miska, E.A., et al. (2005), ‘MicroRNA expression profiles classify human cancers’, *Nature* **435**, 834–838.

Luan, Y. & Li, H. (2003), ‘Clustering of time-course gene expression data using a mixed-effects model with *B*-splines’, *Bioinformatics* **19**, 474–482.

McCulloch, C.E. & Searle, S.R. (2001), *Generalized, Linear, and Mixed Models*, Wiley.

McLachlan, G.J., Do, K.A. & Ambroise, C. (2004), *Analyzing Microarray Gene Expression Data*, Wiley.

Ng, S.K., Krishnan, T. & McLachlan, G.J. (2004), The EM algorithm, *in* J. Gentle, W. Hardle & Y. Mori, eds, 'Handbook of Computational Statistics Vol. 1', Springer-Verlag, pp. 137–168.

Ng, S.K., McLachlan, G.J., Wang, K., Ben-Tovim, L. & Ng, S.-W. (2006), 'A mixture model with random-effects components for clustering correlated gene-expression profiles', *Bioinformatics* **22**, 1745–1752.

Pavlidis, P., Li, Q. & Noble, W.S. (2003), 'The effect of replication on gene expression microarray experiments', *Bioinformatics* **19**, 1620–1627.

Schwarz, G. (1978), 'Estimating the dimension of a model', *Ann. Stat.* **6**, 461–464.

Wang, X., Li, A., Jiang, Z.H. & Feng, H.Q. (2006), 'Missing value estimation for DNA microarray gene expression data by support vector regression imputation and orthogonal coding scheme', *BMC Bioinformatics* **7**, Article 32.

Yeung, K.Y., Fraley, C., Murua, A., Raftery, A.E. & Ruzzo, W.L. (2001), 'Model-based clustering and data transformations for gene expression data', *Bioinformatics* **17**, 977–987.

Yeung, K.Y., Medvedovic, M. & Bumgarner, R.E. (2003), 'Clustering gene-expression data with repeated measurements', *Genome Biol.* **4**, Article R34.

A Maximally Diversified Multiple Decision Tree Algorithm for Microarray Data Classification

Hong Hu[1] **Jiuyong Li**[1] **Hua Wang**[1] **Grant Daggard**[2] **Mingren Shi**[1]

[1]Department of Mathematics and Computing
[2]Department of Biological and Physical Sciences
University of Southern Queensland,
Toowoomba, QLD 4350, Australia
Email: huhong@usq.edu.au

Abstract

We investigate the idea of using diversified multiple trees for Microarray data classification. We propose an algorithm of Maximally Diversified Multiple Trees (MDMT), which makes use of a set of unique trees in the decision committee. We compare MDMT with some well-known ensemble methods, namely AdaBoost, Bagging, and Random Forests. We also compare MDMT with a diversified decision tree algorithm, Cascading and Sharing trees (CS4), which forms the decision committee by using a set of trees with distinct roots. Based on seven Microarray data sets, both MDMT and CS4 are more accurate on average than AdaBoost, Bagging, and Random Forests. Based on a sign test of 95% confidence, both MDMT and CS4 perform better than majority traditional ensemble methods tested. We discuss differences between MDMT and CS4.

Keywords: ensemble classifier, diversified classifiers, decision tree, Microarray data.

1 Introduction

DNA Microarray technology provides capability to monitor the expression levels of thousands of genes at one time. Microarray data analysis offers the potential for discovering the causes of diseases, and identifying the marker genes which might be the signature of certain diseases.

In response to this potential, many Microarray classification algorithms have been proposed in the past ten years. Most of them have been adapted from data mining and machine learning methods, such as support vector machines (SVMs) (Brown, Grundy, Lin, Cristianini, Sugnet, Furey, Jr & Haussler 2000, Guyon, Weston, Barnhill & Vapnik 2002), k-nearest neighbor classifier (Yeang, Ramaswamy, Tamayo & et al. 2001), ensemble methods including Bagging and Boosting (Tan & Gibert 2003, Dietterich 2000), etc. Many researchers have focused their efforts to the study of ensemble decision tree methods (Li & Liu 2003, Tan & Gibert 2003, Dettling 2004, Zhang, Yu & Singer 2003) since they have shown promise to achieve high classification accuracy and its results are very easy to be interpreted.

Ensemble methods combine multiple classifiers (models) built on a set of re-sampled training data sets, or generated from various classification methods on a training data set. This set of classifiers form a decision committee, which classifies future coming samples. The classification of the committee can be simple vote or weighted vote of individual classifiers in the committee. We focuss on ensemble methods of combining multiple classifiers built on a set of re-sampled training data sets. The essence of ensemble methods is to create diversified classifiers in the decision committee. Aggregating decisions from diversified classifiers is an effective way to reduce bias existing in individual trees. However, if classifiers in the committee are not unique, the committee has to be very large to create certain diversity in the committee.

A quick way to create diversity in the decision committee is to include a set of unique trees. This is a motivation of our proposed algorithm. A concern for such a split is that it might break down some attribute combinations or remove some informative genes that are good for classification. However, it is workable for Microarray data. Firstly, a Microarray data set contains a large number of genes, thousands to tens thousands, and this large number of genes can afford for the removal of small number of genes in subsequent trees. Secondly, Microarray data normally contains many noise values. It is very likely that expression levels of some genes are falsely correlated to outcomes (cancer or normal) due to noises. If those genes are repeatedly used in a decision committee, they will cause unreliable predictions in new cases. The diversified trees can avoid such problem. Thirdly, biologists are interested in gene interactions, the use of top genes by information gain ratio may lead to the discovery of trees of few genes. By removing these top genes, more gene combinations may be discovered.

CS4–cascading-and-sharing trees (Li & Liu 2003) is a diversified decision tree ensemble. CS4 selects n top genes and then builds n trees from the roots of n top genes. Apart from the root of the tree is fixed, other level of trees are constructed by using a normal tree construction method. CS4 has been shown achieving higher classification accuracy than Bagging and Boosting. It was reported that CS4 is better than other ensemble decision tree methods for Microarray data analysis. However, apart from the top level genes, other genes in the tree are shared. A number of trees may use some genes repeatedly. Thus, noise from one gene may affect most trees. Also, the performance of CS4 largely replies on the selection of top genes.

A distinction between CS4 and our proposed algorithm is that there are no common genes in our

This project was partially supported by Australian Research Council Discovery Grant DP0559090.

trees in the decision committee whereas genes in trees of CS4 are overlapping except the root genes. We will compare these two diversified decision tree approaches in this paper, and compare them with other traditional ensemble methods.

Complete-random classifiers (Liu, Ting & Fan 2005) also maximize the diversity of ensemble classifiers. Randomly generated trees may overlap, but a large number of trees, for example, thousands to ten thousands, diminish the effect of the overlaps. The results of complete random decision trees are promising too. We do not consider this diversifying approach in this paper based on efficiency consideration.

The rest of this paper is organized as follows. In section 2, we describe the related work on ensemble decision tree classification. In section 3, we introduce our maximally diversified multiple decision tree algorithm (MDMT). In section 4, we show experimental results. In Section 5, we present discussions. In section 6, we conclude the paper.

2 Related work

Bagging, Boosting and Random forests are some well-known ensemble methods in the machine learning field.

Bagging was proposed by Leo Breiman (Breiman 1996) in 1996. Bagging uses a bootstrap technique to re-sample the training data sets. Some samples may appear more than once in a data set whereas some samples do not appear. A set of alternative classifiers are generated from a set of re-sampled data sets. Each classifier will in turn assign a predicted class to an coming test sample. The final predicted class for the sample is determined by the majority vote. All classifiers have equal weights in voting.

Boosting was first developed by Freund and Schapire (Freund & Schapire 1996) in 1996. Boosting uses a re-sampling technique different from Bagging. A new training data set is generated according to its sample distribution. The first classifier is constructed from the original data set where every sample has an equal distribution ratio of 1. In the following training data sets, the distribution ratios are made different among samples. A sample distribution ratio is reduced if the sample has been correctly classified; Otherwise the ratio is kept unchanged. Samples which are misclassified often get duplicates in a re-sampled training data set. In contrast, samples which are correctly classified often do not appear in a re-sampled training data set. A weighted voting method is used in the committee decision. A higher accuracy classifier has larger weight than a lower accuracy classifier. The final verdict goes along with the largest weighted votes.

Tan and Gilbert (Tan & Gibert 2003) used Bagging and Boosting C4.5 decision trees. For Microarray data classification, the results showed that both methods outperform C4.5 single tree on some Microarray cancer data sets. Statistik and Surich developed a new BagBoosting method (Dettling 2004). Their experiments showed that BagBoosting outperforms constantly over Boosting and Bagging methods and achieved a better accuracy result on some Microarray data sets compared with some well-known single classification algorithms such as SVM and kNN.

Zhang and et al. (Zhang et al. 2003) proposed a new ensemble decision tree method called deterministic forest which was a modified version of random forests. Instead of re-sampling the training data set, this method selects a specified number of the top splits of the root node and then generates a number of alternative trees. The accuracy of results from deterministic forests are comparable to random forests.

CS4–cascading-and-sharing proposed by Jinyan Li and Huiqing Liu (Li & Liu 2003) makes use of both in their ensemble C4.5 algorithm for Microarray data classification. CS4 first uses the information gain ratio to select top n genes from the original data set. Then each of n genes in turn is used as the root node of an alternative tree of ensemble trees. Root nodes of ensemble trees are not determined by C4.5, but the remaining parts of trees are constructed by C4.5. CS4 diversifies roots of ensemble decision trees, but does not diversify all trees in the committee as our proposed algorithm.

3 Maximally diversified multiple decision tree algorithm (MDMT)

To improve the accuracy and reliability of ensemble decision tree methods for Microarray classification, we propose a new maximally diversified multiple decision tree (MDMT) method. We avoid the overlapping genes among alternative trees during the tree construction stage. MDMT guarantees that constructed trees are truly unique and maximizes the diversity of the final classifiers. By doing this, MDMT will reduce the instability caused by overlapping genes in current ensemble methods. For example, if the expression level of one gene is read wrongly, it only affects one tree and all other trees are unaffected.

MDMT algorithm consists of the following two steps:

1. Tree construction

   The aim of this step is to construct multiple decision trees by re-sampling genes. All trees are built on all samples but with different sets of genes. We conduct re-sampling in a systematic way. First, all samples with all genes are used to build the first decision tree. After the decision tree is built, the used genes are removed from the data. All samples with remaining genes are used to built the second decision tree. Then the used genes are removed. This process repeats until the number of trees reaches the preset number. As a result, all trees are unique and do not share common genes.

Algorithm 1 Maximally diversified multiple decision tree (MDMT)

```
train($D, \mathcal{T}, n$)
  INPUT: A Microarray data set $D$, and the number of trees $n$.
  OUTPUT: A set of disjointed trees $\mathcal{T}$
  let $\mathcal{T} = \emptyset$
  for $i = 0$ to $n - 1$ do
    call c4.5 to build tree $T_i$ on $D$;
    remove genes used in $T_i$ from $D$;
    $\mathcal{T} = \mathcal{T} \cup T_i$.
  end for
  Output $\mathcal{T}$;
CLASSIFY($\mathcal{T}, x, n$)
  INPUT: A set of trained trees $\mathcal{T}$, a test sample $x$, and the
  number of trees $n$.
  OUTPUT: A class label of $x$
  let vote($i$) = 0 where $i = 1$ to $c$ = the number of classes.
  for $j = 1$ to $n$ do
    let $c$ be the class outputted by $T_j$;
    vote($c$) = vote($c$) + accuracy($T_j$);
  end for
  Output $c$ that maximizes vote($c$);
```

2. Classification

   Since the k-th tree can only use the genes that have not been selected by the previously created k-1 trees, the quality of k-th tree might be decreased. To avoid this problem, The final predicted class of a coming unseen sample is determined by the weighted votes from all trees. Each tree is given the weight of its training classification accuracy rate. The value of each vote is weighted by accuracy of tree making prediction. The majority vote is endorsed as the final predicted class. When the vote is tie, the class predicted by the first tree is advantaged. Since all trees are built on the original data set, all trees are accountable on all samples. This avoids unreliability of voting caused by sampling a small data set. Since all trees make use of different sets of genes, trees are independent. This brings another merit to this diversified committee. One gene containing noise or missing values only affects one tree but not multiple trees. Therefore, it is expected to be reliable in Microarray data classification where noise and missing values prevail.

The complete list of MMDT algorithm is given in Algorithm 1.

We give some explanations of the algorithms in the following.

C4.5 is itself a gene selection algorithm based on information gain ratio. Therefore, no gene selection algorithm is required. In addition, C4.5 discretizes continuous values by information gain ratio. No discretization pre-process is required for this algorithm. The algorithm works on the set of the original data set.

The input is a Microarray data set and a preset number of trees. The first tree (T_1) is constructed based on the original training data set. The second tree (T_2) is based on a re-sampled training data set where genes used in T_1 are removed. As a result, T_1 and T_2 share no common genes and hence are unique. The process repeats until the required number of trees k is generated.

4 Experimental results

To evaluate the performance of ensemble decision tree methods, Seven data sets from Kent Ridge Biological Data Set Repository (Li & Liu 2002) are selected. Table 1 shows the summary of the characters of the seven data sets. We conduct our experiments by using tenfold cross-validation on the merged original training and test data sets.

Table 1: Experimental data set details

Data set	Genes	Class	Record
Breast Cancer	24481	2	97
Lung Cancer	12533	2	181
Lymphoma	4026	2	47
Leukemia	7129	2	72
Colon	2000	2	62
Ovarian	15154	2	253
Prostate	12600	2	21

Our developed MDMT algorithm is compared with five well known single and ensemble decision tree algorithms, namely C4.5, Random Forests, AdaBoostC4.5, Baggingc4.5 and CS4. We have done our experiments with all four algorithms apart from CS4 using the Weka-3-5-2 package which is available online (`http://www.cs.waikato.ac.nz/ml/weka/`). We have done the experiments with CS4 using the software tool provided by Dr Jinyan Li and Huiqing Liu. Default settings are used for all compared ensemble methods. We were aware that the accuracy of some methods on some data sets can be improved when parameters were changed. However, it was difficult to find another uniform settings good for all data sets. Therefore, we did not change default settings since the default produced higher accuracy on average. From our experiments, we found that a large number of ensemble trees does not necessarily improve the prediction accuracy. We use C4.5 default settings for our MDMT algorithm and set the number of trees as 25 for the tenfold cross-validation test since further increasing the number of ensemble trees does not help to improve the prediction accuracy of classification.

Table 2 shows the individual and average accuracy results of the six methods based on tenfold cross-validation method.

Based on tenfold cross-validation test, our MDMT outperforms other ensemble methods. Compared to the single decision tree, MDMT is the best ensemble method and outperforms C4.5 by 10.0% on average. CS4 also performs very well and improve the accuracy on average by 8.4%. Random Forests, Adaboostc4.5 and BaggingC4.5 improves the accuracy on average by up to 4.3%. Among the five ensemble methods, MDMT is the most accurate classification algorithm and improves the accuracy of classification on all cancer data sets by up to 26.7%. CS4 is comparable to MDMT in the test and improves the accuracy of classification on all data sets by up to 17.4%. Baggingc4.5 also outperforms C4.5 on all data sets by up to 9.6%. Random Forests improves the accuracy on lung cancer, Lymphoma, Leukemia and Prostate data sets by up to 19.1%, but fails to improve the accuracy on breast cancer, Colon and Ovarian data sets. AdaBoostc4.5 only improves the accuracy on Lung Cancer,Lymphoma and Leukemia and decreases the accuracy performance on Breast Cancer and Colon data sets.

To determine whether MDMT and CS4 significantly outperform ensemble traditional methods, we also conducted a sign test. The results are shown in Table 3. Based on a sign test of 95% confidence level, MDMT performs better than C4.5, Random Forests, AdaBoostC4.5 and BaggingC4.5. CS4 performs better than Random Forests and AdaBoostC4.5. Not enough evidence supports that CS4 is better than C4.5 and BaggingC4.5. Both MDMT and CS4 do not perform differently based on this test.

5 Discussions

Our experiments show that diversified ensemble classifiers outperform majority traditional ensemble classifiers tested. This suggests that diversity improves classification accuracy of ensemble classification. However, no evidence shows which diversified decision tree method is better between CS4 and MDMT. In this section, we discuss their relative strengths and weaknesses.

CS4 includes a set of decision trees in the decision committee with a set of distinct top genes at roots. The top genes are identified using information gain ratio in current CS4 algorithm. Apparently, other criteria can be used to find top genes too. If top genes are biologically meaningful, this algorithm is very useful for biologists. It groups genes by some informative genes and builds classifier based on meaningful gene groups. However, if the top genes are misidentified due to noise, the classifier committee is misleading. In addition, apart from the top genes, other genes in trees overlap. One noise gene may affect a number of trees.

In MDMT algorithm, a noise gene only affects one tree, and hence the MDMT should tolerate more noise than CS4 does. One concern of MDMT is that the enforcement of unique trees breaks up some gene combinations that are good for classification. However, the experimental results do not indicate that this is

Data set	C4.5	Random Forests	AdaBoostC4.5	BaggingC4.5	CS4	MDMT
Breast Cancer	62.9	61.9	61.9	66.0	68.0	64.3
Lung Cancer	95.0	98.3	96.1	97.2	98.9	98.9
Lymphoma	78.7	80.9	85.1	85.1	91.5	94.1
Leukemia	79.2	86.1	87.5	86.1	98.6	97.5
Colon	82.3	75.8	77.4	82.3	82.3	85.8
Ovarian	95.7	94.1	95.7	97.6	99.2	96.4
Prostate	33.3	52.4	33.3	42.9	47.6	60
Average	75.3	78.5	76.7	79.6	83.7	85.3

Table 2: Average accuracy of seven data sets with six classification algorithms based on tenfold cross-validation

	C4.5	Random Forests	AdaBoostc4.5	Baggingc4.5	CS4	MDMT
MDMT	(7,0,0)	(7,0,0)	(7,0,0)	(5,2,0)	(3,3,1)	–
P-value	0.008	0.008	0.008	0.031	0.313	–
CS4	(6,0,1)	(6,1,0)	(7,0,0)	(6,0,1)	–	(3,3,1)
P-value	0.063	0.016	0.008	0.063	–	0.313

Table 3: Summary of sign test between MDMT and other classification methods. The second row summaries the pairwise comparison (higher, lower, tie) between MDMT and another classification method based on Table 2. The third rows show the P-values of the test. The same test for CS4 is listed in the next two rows.

a case. This does affect finding some combinations of highly informative genes with less informative genes. This is a minus. However, it finds some combinations of less informative genes that are missed by CS4. This is a plus. Keep in mind that many biologists believe that many "uninformative genes" play an important role in diseases. MDMT has potential for finding such genes combinations missed by CS4.

In short, CS4 is capable of finding informative genes and the combinations of informative genes with informative genes, and of informative genes with less informative genes. MDMT is capable of discovering combinations of informative genes with informative genes, and of less informative genes with less informative genes. In addition, MDMT has potential of being less sensitive to noise data than CS4. Note that informative or less informative genes may only make sense to data analyzers. For biologists, two methods use different gene sets and different combinations to equally explain a Microarray data. Both have potential to offer biologists some interesting discovery.

6 Conclusion

In this paper, we studied using diversified multiple decision trees to classify Microarray data. We proposed an algorithm that maximally diversifies trees in the ensemble decision tree committee. Trees in the committee share no common genes. Genes in trees are not randomly selected, but are chosen by C4.5 in a covering-algorithm manner. We conducted experiments on seven Microarray cancer data sets. The experimental results show that the proposed method and another existing diversified decision tree method, which diversifies trees by using distinct tree roots, are more accurate on average than other well-known ensemble methods, such as Bagging, Boosting and Random Forests. A sign test with 95% confidence shows that both diversified algorithms perform better than majority ensemble methods tested. The experiments indicate that diversity improves classification accuracy of ensemble classification on Microarray data. We discussed the relative strengths and weaknesses of both diversified ensemble classification methods.

References

Breiman, L. (1996), 'Bagging predictors', *Machine Learning* **24**(2), 123–140.

Brown, M., Grundy, W., Lin, D., Cristianini, N., Sugnet, C., Furey, T., Jr, M. & Haussler, D. (2000), Knowledge-based analysis of microarray gene expression data by using suport vector machines, *in* 'Proc. Natl. Acad. Sci.', Vol. 97, pp. 262–267.

Dettling, M. (2004), 'Bagboosting for tumor classification with gene expression data', *Bioinformatics* **20**(18), 3583–3593.

Dietterich, T. G. (2000), 'An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization', *Machine learning* **40**, 139–157.

Freund, Y. & Schapire, R. E. (1996), Experiments with a new boosting algorithm, *in* 'International Conference on Machine Learning', pp. 148–156.

Guyon, I., Weston, J., Barnhill, S. & Vapnik, V. (2002), 'Gene selection for cancer classification using support vector machines', *Machine Learning* **46**(1-3), 389–422.

Li, J. & Liu, H. (2002), 'Kent ridge bio-medical data set repository. http://citeseer.ist.psu.edu/liu95chi.html'.

Li, J. & Liu, H. (2003), Ensembles of cascading trees, *in* 'ICDM', pp. 585–588.

Liu, F. T., Ting, K. M. & Fan, W. (2005), Maximizing tree diversity by building complete-random decision trees., *in* 'PAKDD', pp. 605–610.

Tan, A. C. & Gibert, D. (2003), 'Ensemble machine learning on gene expression data for cancer classification', *Applied Bioinformatics* **2**(3), s75–s83.

Yeang, C., Ramaswamy, S., Tamayo, P. & et al. (2001), 'Molecular classification of multiple tumor types', *Bioinformatics* **17**(Suppl 1), 316–322.

Zhang, H., Yu, C.-Y. & Singer, B. (2003), 'Cell and tumor classification using gene expression data: Construction of forests', *Proceeding of the National Academy of Sciences* **100**(7), 4168–4172.

Comparing SVM sequence kernels: A protein subcellular localization theme

Lynne Davis [ab], **John Hawkins** [ab], **Stefan Maetschke** [ab], **Mikael Bodén** [b]

[a]ARC Centre for Complex Systems, [b]School of Information Technology and Electrical Engineering, The University of Queensland, QLD 4072, Australia.

Abstract

Kernel-based machine learning algorithms are versatile tools for biological sequence data analysis. Special sequence kernels can endow Support Vector Machines with biological knowledge to perform accurate classification of diverse sequence data. The kernels relative strengths and weaknesses are difficult to evaluate on single data sets.

We examine a range of recent kernels tailor-made for biological sequence data (including the Spectrum, Mismatch, Wildcard, Substitution, Local Alignment and a new Profile-based Local Alignment kernel) on a range of classification problems (protein localization in bacteria, peroxisomal protein import signals and sub-nuclear localization). The profile-based local alignment kernel ranks highest, but its computational cost is also higher than for any of the other kernels in contention. The kernels that consistently perform well and tend to produce the most distinct classifications are the Local Alignment, Substitution and Mismatch kernels, suggesting that the exploration of new problem sets should start with these three.

1 Introduction

Support Vector Machines (SVMs) have proved effective on a broad range of biological sequence problems. Examples include the detection of remote protein homologues (Jaakkola, Diekhans & Haussler 2000, Leslie, Eskin & Grundy 2002, Saigo, Vert, Ueda & Akutsu 2004, Rangwala & Karypis 2005), prediction of protein subcellular localization (Hua & Sun 2001*a*), prediction of promoter location and their transcription start sites (Gordon, Towsey, Hogan, Mathews & Timms 2006), and classification of protein secondary structure (Hua & Sun 2001*b*) to mention but a few.

The power of SVMs partly stems from their ability to deal with data in high-dimensional (even infinite) feature spaces without compromising generalization to novel samples. The classification boundary is defined in terms of support vectors, selected from a training sample set to maximize a margin of separation between samples of opposite classes in the feature space–a property that alleviates overfitting.

Since nucleotides and amino acids are distinct monomers, biological sequence data is inherently symbolic. However, many machine learning algorithms require samples to be presented as numeric, fixed-length vectors. Consequently, practitioners have come up with problem-specific ways of encoding sequence data and dealing with varying sequence-lengths.

SVMs (together with support vector regression) are examples of so-called kernel methods (Schölkopf & Smola 2002). Perhaps the most intriguing possibility offered by SVMs is that the kernel–which maps samples in pairs to the feature space–is easily replaced. The choice of kernel is essential as it directly affects the separation of samples in the feature space.

Equation 1 illustrates the decision made by SVMs (Schölkopf & Smola 2002).

$$f(\mathbf{x}) = \sum_{i=1}^{n} y_i \alpha_i \mathbf{x_i}^T \mathbf{x} + b \qquad (1)$$

where $y_i \in \{-1, +1\}$ is the target class for sample $i \in \{1, ..., n\}$, $\mathbf{x_i}$ is the vector describing the ith sample and α_i is the ith Lagrange multiplier which is determined by training the SVM. Instead of directly calculating the dot product, a kernel function, $\kappa(\cdot, \cdot)$, is used to evaluate it. With the kernel function in place there is no need to explicitly define the mapping to the feature space. This is known as "the kernel trick" (Schölkopf & Smola 2002). Kernel methods thus supply a principal way to introduce domain-dependent knowledge without requiring a numeric encoding of each sample (Schölkopf & Smola 2002).

A number of sequence-based kernels have been developed recently, primarily targeted to protein classification problems. In this survey we evaluate the performance of the Spectrum kernel (Leslie et al. 2002), the Mismatch kernel (Leslie, Eskin, Cohen, Weston & Noble 2004), the Wildcard kernel (Leslie & Kuang 2004), the Substitution kernel (Leslie & Kuang 2004), the Local Alignment kernel (Saigo et al. 2004) and a Profile-based Local Alignment kernel.

In this study we provide an independent benchmark of these kernels. They are each trained and tested using five-fold cross-validation on three data sets from the multi-faceted domain of protein subcellular localization (outlined in Section 3). We then perform an analysis of their individual and collective performance. We investigate the correlation between the predictions of the kernels to illustrate the differences in the decision boundaries enabled by each.

2 Methods

We use Platt's Sequential Minimal Optimization (SMO) implementation of the SVM (Platt 1999). In the following sections the terms sequence and sample refer to the protein sequence. Let Σ be the amino acid alphabet. The sequence is a string of amino acids, $\mathbf{s} \in \Sigma^{|\mathbf{s}|}$. The term k-mer similarly refers to k consecutive amino acids, $\alpha = \alpha_1, \alpha_2, ..., \alpha_k \in \Sigma^k$.

Copyright ©2006, Australian Computer Society, Inc. This paper appeared at The 2006 Workshop on Intelligent Systems for Bioinformatics (WISB2006), Hobart, Australia. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 73. Mikael Bodén and Timothy L. Bailey, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

2.1 Spectrum Kernel

For a given sequence, the *spectrum* of a sequence involves all k-mers it contains. The Spectrum feature map is

$$\Phi_k^{spctrm}(\mathbf{s}) = \phi_\alpha(\mathbf{s})_{\alpha \in \Sigma^k} \tag{2}$$

where $\phi_\alpha(\mathbf{s})$ is the simple count of occurrences of α in the sequence $\mathbf{s}$. The Spectrum kernel then compares any two sequences by considering the number of these k-mers that two sequences share (Leslie et al. 2002). More specifically, the kernel calculates the dot product between the vectors holding all k-mer counts for any pair of sequences.

$$\kappa_k^{spctrm}(\mathbf{s_1}, \mathbf{s_2}) = \langle \Phi_k^{spctrm}(\mathbf{s_1}), \Phi_k^{spctrm}(\mathbf{s_2}) \rangle \tag{3}$$

If two sequences share a large number of k-mers their product is large. An important feature of the Spectrum kernel is that it disregards the position of the k-mers within the sequence. Thus, for small values of k, information about the order of the amino acids within the sequence is lost.

2.2 Mismatch Kernel

The Mismatch kernel (Leslie et al. 2004) extends the Spectrum kernel, still tracking the number of k length segments shared by the sequences, but allowing a specified number of mismatches m by which the k-mers can differ. More specifically, the Mismatch feature map is

$$\Phi_{k,m}^{msmtch}(\mathbf{s}) = \sum_{\alpha \in \mathbf{s}} \phi_\beta(\alpha)_{\beta \in \Sigma^k} \tag{4}$$

where all possible α k-mers in $\mathbf{s}$ are expanded to all β k-mers within a certain neighborhood $N_m^{msmtch}(\alpha)$ (includes all k-mers differing by no more than m mismatches from α ignoring position). $\phi_\beta(\alpha) = 1$ if β belongs to $N(\alpha)$, $\phi_\beta(\alpha) = 0$ otherwise.

The kernel result is the dot product between the two k-mer count vectors (as with the Spectrum kernel). If $m = 0$ the Mismatch kernel generates identical results to the Spectrum kernel.

2.3 Substitution Kernel

Instead of allowing residues to be replaced by *any* other possible residue as in the Mismatch kernel, the Substitution kernel uses a substitution matrix, S, to compute the pair-wise alignment scores between the two sequences being compared (Leslie & Kuang 2004). Hence, we define another neighborhood $N_{S,k,\sigma}^{subst}(\alpha)$ that includes all β k-mers that fall above a substitution score threshold σ when aligned with α. Note that number and position of mismatches are considered only indirectly through the alignment score.

As with the Mismatch kernel, the kernel simply counts the number of matching k-mers and returns the dot product between the two feature vectors.

2.4 Wildcard Kernel

Unlike the Mismatch kernel and the Substitution kernel, the Wildcard kernel only allows mismatches at specified locations within the k-mer (Leslie & Kuang 2004).

With the Wildcard kernel, the default alphabet is extended with a wildcard character, $\Sigma \cup \{*\}$. The wildcard character matches any amino acid (as '.' does in a regular expression). The presence of the wildcard character in an α k-mer is position-specific, making the matching of β k-mers less permissive than with the Mismatch and Substitution kernels. x is a parameter controlling the number of wildcards that occur in the k-mer.

It was initially thought that the performance of the Mismatch and Wildcard kernels would be very similar. However, preliminary trials suggested otherwise for specific values of k and x. We therefore included both kernels in the study.

2.5 Local Alignment Kernel

The Local Alignment kernel compares two sequences by exploring their alignments (Saigo et al. 2004). An alignment between the two sequences is quantified using an amino acid substitution matrix, S, and a gap penalty setting, g (involving a gap opening penalty imposed every time a gap needs to be created in the sequence and a gap extension penalty imposed for each extension of the gap required to improve the alignment). A further parameter, β, controls the contribution of non-optimal alignments to the final score. Let $\Pi(\mathbf{s_1}, \mathbf{s_2})$ be the set of all possible alignments between sequences $\mathbf{s_1}$ and $\mathbf{s_2}$. The kernel can be expressed in terms of alignment-specific scores, $\varsigma_{S,g}$ (for details of this function see Saigo et al., 2004).

$$\kappa_\beta^{LA}(\mathbf{s_1}, \mathbf{s_2}) = \sum_{\pi \in \Pi(\mathbf{s_1}, \mathbf{s_2})} exp(\beta \varsigma_{S,g}(\mathbf{s_1}, \mathbf{s_2}, \pi)) \tag{5}$$

The benchmark tests were conducted using a ported version of Saigo and colleagues' source code (Saigo et al. 2004).[1]

2.6 Profile Local Alignment Kernel

Evidence is mounting that so-called position-specific substitution matrices (PSSMs; a.k.a. "profiles") disclose important evolutionary information tied to each residue of proteins (Rangwala & Karypis 2005, Kuang, Ie, Wang, Wang, Siddiqi, Freund & Leslie 2005). We adapt the alignment-specific function, ς, in the Local Alignment kernel to use such substitution scores generated by PSI-Blast (max three iterations, E-value threshold is 0.001, using Genbank's non-redundant protein set) in place of the generic substitution matrix, S. Specifically, we define the substitution score as the average of the PSSM-entries for the two sequences (where the entry coordinates are determined from the sequence position of one sequence and the symbol of the other). All other settings are as for the Local Alignment kernel.

There are several alternative ways of exploiting PSSM scores in a kernel setting (Rangwala & Karypis 2005, Kuang et al. 2005) that we are unable to explore here.

3 Case Problems and Materials

Each kernel is tested and evaluated on data sets that relate to protein subcellular localization. The cell is a decentralized but still carefully controlled device, shuttling gene products, like proteins, to various locations where they perform their functions. Mechanisms for this protein traffic control are not yet fully understood and machine learning techniques are being utilized to assist biologists by predicting localization on the basis of protein sequence. These in-silico models can be used to automatically annotate the

[1] To eschew the documented problem of diagonal dominance in the LA kernel matrix, we use the logarithm of each entry as proposed by Saigo and colleagues.

growing number of sequences that are yet to be experimentally characterized (Nakai 2000). The problem of subcellular localization is multi-faceted and thus represents a range of machine learning problems while entertaining a common application theme.

3.1 Problem 1: Localization in Gram-negative bacteria

In simple prokaryotes, there are only a few protein destinations. Lacking a nucleus, proteins are both encoded and translated in the cytoplasm. If they contain an N-terminal signal peptide they will associate with the inner membrane for further translocation and possible secretion. If not, they will simply remain in the cytoplasm. Specifically, in Gram-negative bacteria, there are five destinations. Cytoplasm, inner membrane, outer membrane, periplasm (space between membranes) and extracellular are the target classes for a classifier.

A number of models have been developed for predicting the localization of proteins in Gram-negative bacteria (Gardy, Spencer, Wang, Ester, Tusnady, Simon, Hua, deFays, Lambert, Nakai & Brinkman 2003, Park & Kanehisa 2003, Wang, Sung, Krishnan & Li 2005). The most recent makes use of a cleverly designed sequence encoding and SVMs (Wang et al. 2005). Recent efforts have highlighted several intricate details underpinning the dynamic process of inserting a protein into the membrane (White & von Heijne 2005). However, as we wish to benchmark a variety of kernels against one another, we refrain from making experimental observations explicit in the simulation design. We use the same data set as in these previous studies, compiled by Gardy et al (Gardy et al. 2003) taken from Swiss Prot release 40.29. This data set contains 1572 protein sequences separated into five subcellular localizations. Of these we use the 1408 that have a single subcellular location and no unknown residues (numbers per class shown in Table 1).

3.2 Problem 2: Peroxisomal targeting

In eukaryotic cells, the complexity of protein localization is much greater. Like prokaryotes, targeting to the secretory pathway is effected by an N-terminal signal peptide as it emerges from the ribosome. The process occurs in tandem with translation, and thus dominates many of the other targets, e.g. the small peroxisome. Peroxisomal proteins are recognized and imported after synthesis in the cytoplasm and targeting is believed to rely on a small number of sequence patterns. The dominating targeting signal is known as PTS1 and appears at the C-terminus. The PTS1 consists of a strongly conserved tri-peptide but several dependencies and constraints range a larger region exposed to the chaperone that play a central role in import (Neuberger, Maurer-Stroh, Eisenhaber, Hartig & Eisenhaber 2003). Previous approaches have employed intricate pre-filtering and constrained encodings of sequence data on basis of experimental observations (Emanuelsson, Elofsson, von Heijne & Cristobal 2003). Again, we refrain from including such constraints to allow a fair comparison between the different kernel functions.

Differentiating between PTS1 targeted peroxisomal proteins and all others with a similar C-terminal signature, constitutes test problem two. The data set contains 124 positive examples and 182 negative examples extracted from Swiss Prot release 45 (Hawkins & Bodén 2005).

3.3 Problem 3: Sub-nuclear Localization

A significant portion of proteins in the eukaryotic cell are shuttled into the nucleus where they can fulfill various regulatory roles. Within the nucleus, proteins tend to concentrate in certain functional areas even though such areas are not physically contained by a membrane. Some proteins are also shuttled back to the cytoplasm. Differentiating between sub-nuclear locations represents yet another angle on the localization problem. As test problem three, we use a data set that distinguishes between six sub-nuclear destinations (Lei & Dai 2005) extracted from the Nuclear Protein Database (NPD) (Dellaire, Farrall & Bickmore 2003). This data set contains 598 proteins in total, 504 separated into six localizations, and 92 with multiple localizations. Again only the singularly localized proteins were used (numbers per class shown in Table 3). One recent study demonstrated the accuracy of an SVM on this task using a tailor-made kernel (Lei & Dai 2005). We investigate how generic kernels perform on this specific problem.

4 Algorithms

4.1 Performance Measures

The kernels were tested on the their ability to assist the SVMs to correctly classify proteins. The SVM predicts a label for each sequence sample $\mathbf{s}$ in the sample space, by $f(\mathbf{s}) \in \{+1, -1\}$. If $f(\mathbf{s_i}) = y_i$ the ith classification is *true*, otherwise it is *false*. If $f(\mathbf{s_i}) = +1$ the prediction is *positive* otherwise *negative*.

To provide a good estimate of the expected prediction accuracy on novel samples, we use five-fold cross validation. All available samples are randomly distributed into five approximately equal and mutually exclusive sets. By training five models on different combinations of four of the five subsets we can assess the *test* accuracy of each subset exactly once. For each class c, we determine the number of *true positives*, tp_c, *true negatives*, tn_c, *false positives*, fp_c, and *false negatives*, fn_c.

The comparison of the kernels is based on two performance measures. We use the accuracy of prediction as a measure that is sensitive to differences in the class distribution. It is defined as the percentage of *positive samples* that are correctly classified.

$$acc(c) = \frac{tp_c}{tp_c + fn_c} \cdot 100 \qquad (6)$$

In contrast, we also report the (Matthews) correlation coefficient, $r(c)$ (Matthews 1975) as a measure that adjusts for imbalances in the class distribution (see Equation 7). $r(c)$ equals $+1$ if the the observations and predictions of members of c are in perfect agreement, -1 if they are in perfect disagreement and 0 if predictions are random.

$$r(c) = \frac{tp_c tn_c - fp_c fn_c}{\sqrt{(tp_c + fn_c)(tp_c + fp_c)(tn_c + fp_c)(tn_c + fn_c)}}. \qquad (7)$$

The *overall* accuracy is defined as

$$acc = \frac{\sum_c tp_c}{N} \cdot 100 \qquad (8)$$

where N is the total number of samples, while the *overall* correlation coefficient r is the *average* of class-specific $r(c)$.

4.2 Architectures

SVMs are inherently binary classifiers. Thus, for multi-class problems we use several SVMs and combine them. The Gram-negative bacteria data set was evaluated using the one-versus-all combination strategy, where each class is allocated a SVM that is trained with all samples from the class making up the positive set and all samples from other classes combined to make the negative set. The predicted class of the ensemble corresponds to the SVM with the highest output as given by Equation 1).

The peroxisomal PTS1 targeting data consists of only positive and negative data, making a single binary classifier possible. Following Lei and Dai (Lei & Dai 2005), the one-versus-one strategy was used to evaluate the sub-nuclear data. For the classification of the six classes, we require a classifier for each distinct pair of classes within the 6, $\mathcal{C}(6, 2) = 15$ (a. k. a. '6 choose 2'). The prediction was based on a jury voting system, in which the sequence was classified to be of the class with which the most classifiers identified. In the case of a tie, the sequence was classified to belong to the class for which the sum of Equation 1 was the greatest.

4.3 Kernel Parameters

In preliminary trials with the Spectrum, Mismatch and Wildcard kernels it was observed that the performance deteriorates when k is greater than 5. Therefore simulations were carried out with k-values ranging from 2 to 5. For the Mismatch and Wildcard kernels m (or x) was limited to 1 and 2, as it was observed in preliminary trials that performance greatly deteriorates for values greater than this (for values of k in the given range). All possible combinations (within the aforementioned boundaries such that $k \geq m + 1$) were tested for these three kernels.

During preliminary trials of the Substitution kernel it was observed that with $\sigma = -1$, using a BLOSUM-62 matrix, and $k = 3$ generated the best result. All the experiments reported herein were done using these settings. Previous studies (Leslie & Kuang 2004) showed that the performance of a Substitution kernel seems stable as k is varied while σ is adjusted additively.

The tests conducted with the Local Alignment kernel (and the Profile Local Alignment kernel) used the same parameter settings used by Saigo and colleagues (Saigo et al. 2004), namely a gap opening penalty of 12 and gap extension penalty of 2. Preliminary trials found that changing the values for the gap opening and extension penalties had only minor effect on the result. Preliminary tests also agreed with Saigo and colleagues finding a β value of 0.5 to be optimal over the range of trials. Hence detailed exploration into the effects of variation in these parameters was not pursued. The use of different substitution matrices was not explored for the Local Alignment kernel to keep consistency across the kernels, only the BLOSUM-62 matrix was used.

5 Results

The performance results for each of the kernels on each of the problems are tabulated. The results for Problem One are shown in Table 1, Problem Two in Table 2 and Problem Three in Table 3. The results displayed show only the best correlation coefficient achieved for each kernel, over the range of parameters explored.

5.1 Localization in Gram-negative bacteria

If we average the correlation coefficient across all the classes of the Problem set 1, shown in Table 1, the alignment-based kernels outperformed all of the spectrum-based kernels. Of the five different localizations in the data set, both alignment-based kernels had better $r(c)$ than the spectrum-based kernels for four of them (only inferior for the localization of the inner membrane proteins). The Profile Local Alignment kernel was outstanding overall.

Of the spectrum-based kernels, the Mismatch and Wildcard variants performed best, with almost identical correlation coefficients. The similarity in their performance is not surprising, however the parameters used to get the optimal results for each are slightly different. The Mismatch kernel performed best with $k = 4$ and $m = 1$, whereas the Wildcard kernel performed best with $k = 4$ and $x = 2$. These results highlight the difference between these two kernels; in the Mismatch kernel the location of the mismatch in the k-mer is not taken into account, whereas in the Wildcard kernel it is. There is a larger space for error (i.e. matching two k-mers that are not related) in the Mismatch kernel, particularly for larger values of m.

The Substitution kernel finds sequence similarities by separating the sequence into all possible spectrums, and comparing the spectrums using a substitution matrix to allow some flexibility in amino acid composition. In the present work the BLOSUM-62 matrix was used with reasonable utility by the Substitution kernel. A different substitution matrix could potentially accommodate the problem domain more effectively, e.g. to readily accept substitutions between hydrophobic residues in membrane domains and between Pro and Gly (both serving to break helices).

Previous studies of Gram-negative bacteria protein localization have made use of spectrum-like kernels along with techniques such as amino acid subalphabets (Wang et al. 2005) to achieve very accurate results, reporting a correlation coefficient of 0.874. The present study found that a simple adaptation of the standard Local Alignment kernel ($r = 0.873$) performs just as well.

5.2 PTS1 Peroxisomal Targeting

On the basis of the correlation coefficient the identification of the presence of a PTS1 signal is best performed using an alignment kernel. All other kernels are significantly inferior. One possible explanation for the inferior result of the spectrum-based kernels is that the targeting signal of peroxisomal proteins is known to occur at a specific position. The spectrum-based kernels take information from the whole sequence, creating a spectrum of all k-mers, without regard to position. However, the Local Alignment kernel finds strong alignments between the sequences, which can be at a specific part of the sequence. A high score can thus be based on the part of the sequence corresponding to the location of the signal.

With $r = 0.783$ the Local Alignment kernel is promising. The profile-based kernel outperformed it slightly at $r = 0.797$. The current best performing model in the literature is PTS1Prowler (Hawkins & Bodén 2005) estimated to have a correlation coefficient of 0.766 with a standard deviation of 0.02 (calculated from five training repeats). The present results were produced from only a single cross validation run.

Kernel		Spectrum		Mismatch		Wildcard		Substitution		LA		Profile LA	
Class	#Proteins	Acc	r	Acc	r	Acc	r	Acc	r	Acc	r	Acc	r
Cytoplasm	275	86.6	0.756	90.2	0.790	84.7	0.778	84.0	0.767	92.0	0.838	89.4	0.847
Secreted	190	65.3	0.696	68.4	0.755	71.6	0.758	66.8	0.659	75.8	0.805	84.2	0.848
Inner Membrane	292	88.0	0.884	89.0	0.914	89.4	0.908	88.7	0.870	88.7	0.890	91.1	0.908
Outer Membrane	375	89.9	0.847	92.8	0.893	93.3	0.890	89.3	0.860	94.4	0.906	95.7	0.940
Periplasm	276	76.8	0.702	82.2	0.746	82.6	0.738	74.3	0.664	84.8	0.801	87.3	0.824
Overall	1408	83.0	0.777	86.2	0.820	85.8	0.814	82.2	0.764	88.4	0.848	90.3	0.873

Table 1: **Gram-negative Bacterial Protein Localization**. Comparison of results of the kernels when tested on the Gram-negative bacteria localization problem set. Accuracy and correlation coefficients are given. The kernel parameters for the variable kernels were: Spectrum $k = 3$; Mismatch $k = 4$, $m = 1$; Wildcard $k = 4$, $x = 2$ for all localizations.

Kernel		Spectrum			Mismatch				Wildcard				Substitution		LA		Profile LA	
Class	#Proteins	k	Acc	r	k	m	Acc	r	k	x	Acc	r	Acc	r	Acc	r	Acc	r
PTS1	124/182	3	77.4	0.577	4	1	71.8	0.557	4	2	75.0	0.586	82.3	0.605	81.4	0.783	90.2	0.797

Table 2: **PTS1 Peroxisomal Protein Localization**. Comparison of results of the kernels when tested on the Peroxisomal Targeting Signal problem set. Accuracy and correlations coefficients are given. The kernel parameters for the first three spectrum based kernels are shown prior to the results.

5.3 Sub-nuclear Localization

The more difficult problem of sub-nuclear localization yielded varied results for each of the kernels. Firstly, the best performing parameters for each kernel varied over the different localizations. For the Spectrum, Wildcard and Mismatch kernels, the best k values ranged between two and five, the entire scope of the values explored. Again only one configuration was trialed for both the Local Alignment and Substitution kernels. The variation in optimal parameters for the spectrum-based kernels suggests that sub-nuclear targeting relies on sequence features specific to each location.

The best performing kernels for this data set were the Mismatch kernel, and the Profile Local Alignment kernel. On the basis of the correlation coefficient the standard Local Alignment kernel performed worst of all. If we look at the accuracy we note that the Local Alignment kernel has made a strong deference to the majority class (Nucleolus).

If we presume that this data set is representative and reasonably clean, then it is noteworthy that none of the kernels are able to project the sequence data to a feature space that allows classification to occur reliably. However, with some classes heavily under-represented in the data set, the current problems may dissolve as more data becomes available.

The only existing predictor of sub-nuclear localization (Lei & Dai 2005) makes use of spectrum-based kernels in conjunction with evolutionary information to classify the proteins. Lei and Dai studied a number of different encodings of different spectrum length, with or without evolutionary information. Their best performing predictor combined a number of encodings of different spectrum lengths to achieve a correlation coefficient of 0.284. Although this is higher than any of the results achieved in this study, it is interesting to note that the Mismatch kernel performs comparably to each of the individual components used in the composite model presented by Lei and Dai (Lei & Dai 2005).

5.4 Kernel computation

Besides accuracy, kernels can be evaluated in terms of their computational efficiency. We measured the average duration of computation for all kernels with the aim of supplying further insights into the impact they may have on model training and testing time. To evaluate the scaling of computational time in relation to the length of the sequences we identified three sets of ten non-redundant proteins, each set containing only proteins within a particular size range. The sizes were (1) less than 200 residues, (2) more than 200 but less than 400 residues, and (3) more than 400 residues. The groups had average residue counts of 106, 278, and 478, respectively. We timed the kernel-function calls for each possible pair within each group on a standard PC (2GHz, 1GB RAM, Windows XP/Java) and repeated this procedure five times, averaging the totals, to determine a typical call-duration. Table 4 shows, for the three sub-sets, the call-duration for each kernel with parameter settings used in our study. Durations should be interpreted with caution as they are dependent on implementational details. However, our measurements provide reasonable guidance for determining the extent of training and testing time required. We have excluded the profile-based kernel as it runs PSI-Blast as a pre-processing stage, greatly contributing to the computation time. As a guide to its computational cost, once the PSSM has been determined (which can take several minutes for a single protein), the Profile Local Alignment kernel equals the standard Local Alignment kernel.

As seen in Table 4, the Local Alignment kernel is computationally more expensive than most other kernels that are competitive in terms of accuracy. Other notable offenders include the Mismatch kernel with $m > 1$, and the Substitution kernel with $k > 2$. However, neither of these configurations achieved high accuracy.

6 Analysis

The model with the highest average correlation coefficient across all the problems is the Profile Local Alignment kernel. On a case by case basis this observation is somewhat deceptive. The variation in observed performance indicates that choosing a kernel, even within a mildly constrained problem area such as subcellular localization, should be done on a case by case basis. Nevertheless, a systematic study of the differences between the kernels across these problems reveals certain trends that suggest heuristics for testing kernels on new problems.

The six kernels were compared pairwise to provide further insights into their characteristics. Similar to the calculation of the correlation coefficient between the target and the predicted classifications in a data set, Equation 7 is used to compute a correlation be-

Kernel		Spectrum			Mismatch				Wildcard				Substitution		LA		Profile LA	
Class	#	k	Acc	r	k	m	Acc	r	k	x	Acc	r	Acc	r	Acc	r	Acc	r
PML Body	38	2	7.9	0.021	3	1	15.8	0.137	3	1	13.2	0.134	28.9	0.120	0.0	-0.013	2.6	0.024
Nucleolus	219	3	87.2	0.240	3	1	71.7	0.312	4	2	82.6	0.326	63.0	0.361	90.4	0.260	87.1	0.346
Nucleoplasm	75	5	14.7	0.104	5	1	12.0	0.181	4	1	16.0	0.120	26.7	0.135	13.3	0.061	28.0	0.207
Speckles	56	4	17.9	0.324	4	1	44.4	0.491	4	2	30.4	0.348	18.0	0.118	11.5	0.186	23.2	0.265
Lamina	55	2	34.6	0.326	2	1	32.7	0.303	5	2	18.2	0.265	25.4	0.167	18.2	0.175	27.9	0.381
Chromatin	61	3	11.5	0.123	4	1	13.1	0.166	4	2	19.7	0.197	19.6	0.145	14.3	0.214	21.9	0.210
Overall	504	3	46.6	0.173	4	1	49.0	0.238	4	2	47.8	0.211	40.7	0.174	46.2	0.147	50.4	0.239

Table 3: **Sub-nuclear Protein Localization**. Comparison of results of the kernels when tested on the Subnuclear localization problem set. Accuracy and correlation coefficients are given. The kernel parameters for the different localizations are listed.

		Protein length		
Kernel	Parameters	Short	Medium	Long
LA	$\beta = 0.5$	4.45	27.05	78.17
Spectrum	$k = 1$	0.18	0.15	0.22
Spectrum	$k = 2$	0.07	0.07	0.25
Spectrum	$k = 3$	0.04	0.15	0.18
Spectrum	$k = 4$	0.04	0.15	0.18
Spectrum	$k = 5$	0.04	0.15	0.25
Mismatch	$k = 2, m = 1$	0.95	2.33	3.83
Mismatch	$k = 3, m = 1$	1.49	4.08	6.34
Mismatch	$k = 4, m = 1$	2.73	6.34	10.52
Mismatch	$k = 5, m = 1$	4.22	8.26	14.13
Mismatch	$k = 3, m = 2$	47.63	118.01	201.36
Mismatch	$k = 4, m = 2$	158.39	373.01	635.15
Mismatch	$k = 5, m = 2$	367.51	996.97	1773.02
Wildcard	$k = 2, x = 1$	0.15	0.18	1.17
Wildcard	$k = 3, x = 1$	0.11	0.22	0.36
Wildcard	$k = 4, x = 1$	0.15	0.29	0.77
Wildcard	$k = 5, x = 1$	0.37	0.62	0.80
Wildcard	$k = 3, x = 2$	0.19	0.40	0.73
Wildcard	$k = 4, x = 2$	0.33	1.02	1.64
Wildcard	$k = 5, x = 2$	0.51	1.27	2.18
Substitution	$k = 2, \sigma = -1$	10.40	11.44	12.85
Substitution	$k = 3, \sigma = -1$	171.44	215.27	231.47

Table 4: The average time in milliseconds for each call to a specific kernel-function configured with specific parameter values. Tested data sets contain proteins with less than 200 residues (short), with more than 200, and less than 400 residues (medium) and with more than 400 residues (long).

	Problem			
Kernels	1	2	3	Average
Mismatch - Wildcard	0.93	0.86	0.70	0.83
Spectrum - Mismatch	0.89	0.87	0.61	0.79
LA - Profile LA	0.87	0.79	0.62	0.76
Spectrum - Wildcard	0.85	0.75	0.59	0.73
Mismatch - LA	0.84	0.67	0.59	0.70
Wildcard - Subst	0.82	0.68	0.59	0.69
Wildcard - LA	0.83	0.67	0.57	0.69
Wildcard - Profile LA	0.81	0.67	0.55	0.68
Mismatch - Profile LA	0.81	0.65	0.56	0.67
Mismatch - Subst	0.82	0.63	0.53	0.66
Spectrum - Subst	0.78	0.59	0.56	0.64
Spectrum - LA	0.78	0.61	0.53	0.64
Subst - LA	0.77	0.65	0.51	0.64
Spectrum - Profile LA	0.77	0.59	0.49	0.62
Subst - Profile LA	0.75	0.60	0.48	0.61

Table 5: Correlation coefficients of predicted classifications from pairs of kernels. A correlation of 1 indicates that kernels enable the same predictions. A correlation of 0 indicates that there is chance agreement between predictions.

tween the predictions of two kernels. One kernel is chosen as a reference point. Whenever the other kernel produces the same positive predictions then these are considered true positives, if the second kernel produces a negative prediction where the first produces a positive, it is considered a false negative, and so on. The resulting pairwise correlations between the outputs of kernels can be found in Table 5.

The highest correlating kernels are the Wildcard and Mismatch kernels, which seem to share more predictions than any of the other pairs of kernels. Although the parameters used by the Wildcard and Mismatch kernels are different, the correlation is to be expected due to the similar tactics they employ. The kernels whose prediction is least correlated are the Substitution and Local Alignment kernels. Additionally, they both correlated weakly with each of the other kernels, in particular with the Spectrum kernel.

To investigate the qualititative nature of the feature spaces, we performed Kernel Principal Components Analysis (kernel-PCA) (Schölkopf, Smola & Müller 1999) on Problem set 1. In Figure 1, 10 inner membrane and 10 outer membrane proteins (arbitrarily selected from those subsets) are shown. Specifically, the samples are mapped onto the two dimensions with the largest eigenvalues in the Spectrum kernel $k = 3$ feature space and the Local Alignment kernel feature space, respectively. Kernel-PCA had access to all inner and outer membrane proteins.

From Figure 1, we note that several samples are mapped differently to the feature space, e.g. Q51397 and Q55293 are quite distinct according to the Spectrum kernel but similar according to the Local Alignment kernel. The outer membrane protein Q51922 is misclassified by both kernels but with different outcomes ("cytoplasm" for the Spectrum and "periplasm" for the Local Alignment kernel). The inner membrane protein Q52788 is confused for an outer membrane protein by both kernels (clearly occupying a space in the wrong feature space territory).

7 Conclusion

This paper takes a range of popular sequence kernels and compares their performance over a range of protein subcellular localization problems. Where the content of this study overlaps with previous comparative simulations we are in general agreement. Leslie and colleagues (Leslie et al. 2004) found that adding mismatches to spectrums improves the result on spectrums alone for protein remote homology classification. Furthermore, Saigo and colleagues (Saigo et al. 2004) found that the local alignment kernel outperforms the Mismatch kernel, again on the remote homology problem. Cheng and colleagues (Cheng, Saigo & Baldi 2006) also noted that the Local Alignment kernels outperformed both Mismatch and the Spectrum kernels on a protein disulphide bond detection problem set. Very recent developments indicate the potential of incorporating substitution profiles in the kernels (Rangwala & Karypis 2005, Kuang et al. 2005). We adapt the Local Alignment kernel to use such scores and also find that accuracy improves considerably.

Although the overall performance of the kernels agrees with these results, the performance of the kernels was not consistent across the range of problems. These results demonstrate that when choosing kernels for specific problems, a range of kernels should be considered to ensure the most appropriate ker-

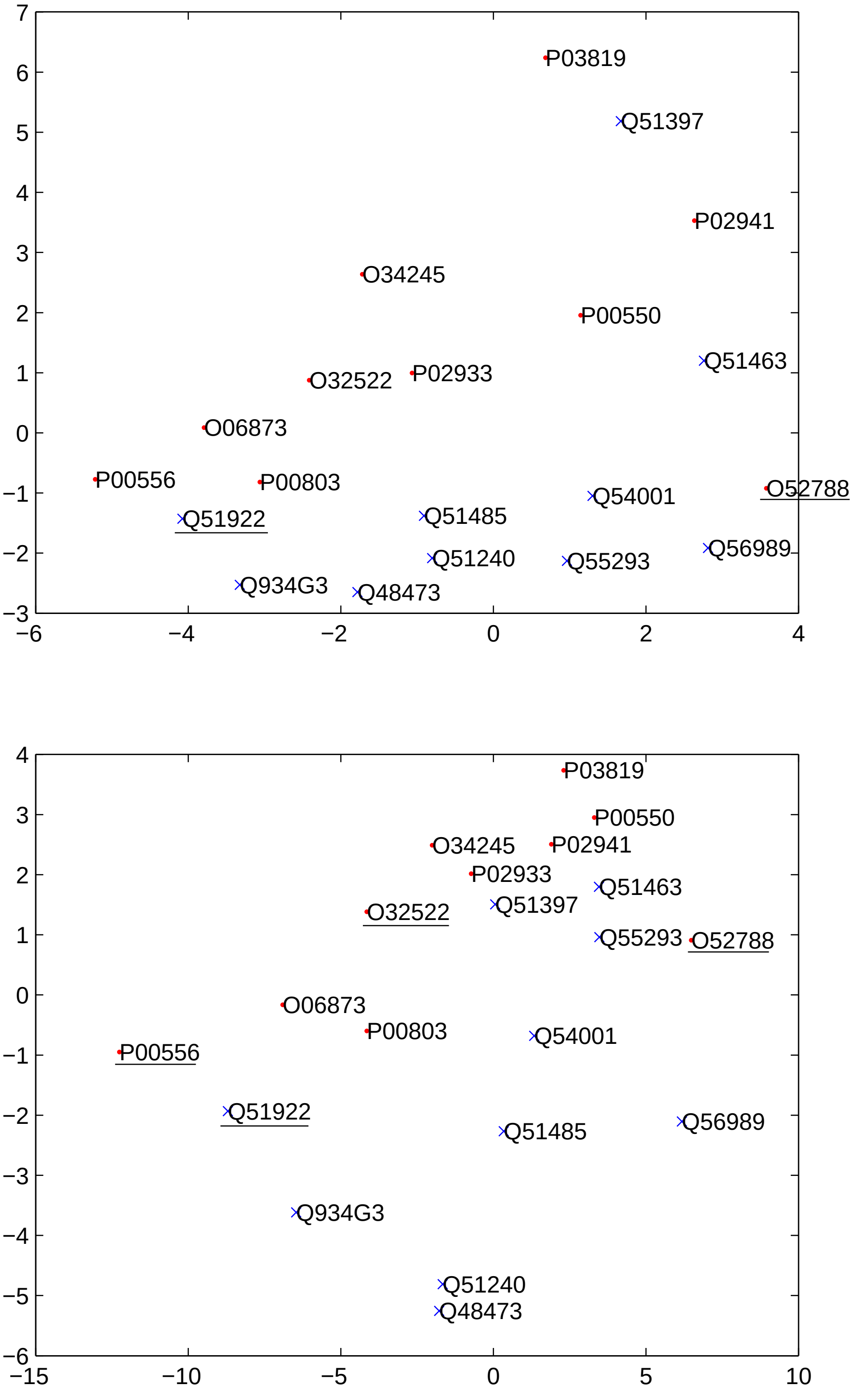


Figure 1: Kernel Principal Component Analysis was performed on the Spectrum kernel $k = 3$ features space (above) and the Local Alignment feature space (below) using Problem set 1 (inner and outer membrane). The same samples are shown in both feature spaces. Each sample is labelled with its Swiss Prot identifier. Inner membrane proteins are plotted as red dots, outer membrane proteins are plotted as blue crosses. Samples that were misclassified in the reported simulations are underlined.

nel is chosen. The correlation between the predictions indicates that the Spectrum, Local Alignment and Substitution kernels are the most distinct methods for mapping sequences to a SVM feature space. However, the spectrum-based Mismatch kernel consistently outperforms the Spectrum kernel and can be easily substituted in its place. Suggesting that the ideal initial experiment should involve the Mismatch, Local Alignment and Substitution kernels to determine the kernel architecture to which the specific problem is suited.

Comparing the kernels in terms of time consistency and efficiency, the Mismatch, Local Alignment and Substitution kernels perform worst. This illustrates that when it comes to choosing a kernel the trade-off between accuracy, correlation of errors and time efficiency can not be avoided with the reviewed range of kernels.

Finally, in our benchmark on the sub-nuclear localization data set, none of the kernels performed satisfactorily. If we presume that this is not due to problems with the data, then we must conclude that the tested range of sequence kernels does not yet offer a complete toolkit for biological sequence classification.

Acknowledgments

This research was supported by the Australian Research Council Centre for Complex Systems and Centre for Bioinformatics. We thank Johnson Shih who implemented some of the kernels used in this study.

References

Cheng, J., Saigo, H. & Baldi, P. (2006), 'Large-scale prediction of disulphide bridges using kernel methods, two-dimensional recursive neural networks, and weighted graph matching', *Proteins: Structure, Function, and Bioinformatics* **62**(3), 617–629.

Dellaire, G., Farrall, R. & Bickmore, W. (2003), 'The nuclear protein database (npd): sub-nuclear localisation and functional annotation of the nuclear proteome', *Nucl. Acids Res.* **31**(1), 328–330.

Emanuelsson, O., Elofsson, A., von Heijne, G. & Cristobal, S. (2003), 'In silico prediction of the peroxisomal proteome in fungi, plants and animals', *Journal of Molecular Biology* **330**(2), 443–456.

Gardy, J., Spencer, C., Wang, K., Ester, M., Tusnady, G., Simon, I., Hua, S., deFays, K., Lambert, C., Nakai, K. & Brinkman, F. (2003), 'PSORT-B: improving protein subcellular localization prediction for gram-negative bacteria', *Nucl. Acids Res.* **31**(13), 3613–3617.

Gordon, J. J., Towsey, M. W., Hogan, J. M., Mathews, S. A. & Timms, P. (2006), 'Improved prediction of bacterial transcription start sites', *Bioinformatics* **22**(2), 142–148.

Hawkins, J. & Bodén, M. (2005), Predicting peroxisomal proteins, *in* 'Proceedings of the IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology', IEEE, Piscataway, pp. 469–474.

Hua, S. J. & Sun, Z. R. (2001*a*), 'Support vector machine approach for protein subcellular localization prediction', *Bioinformatics* **17**(8), 721–728.

Hua, S. & Sun, Z. (2001*b*), 'A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach,', *Journal of Molecular Biology* **308**(2), 397–407.

Jaakkola, T., Diekhans, M. & Haussler, D. (2000), 'A discriminative framework for detecting remote protein homologies', *Journal of Computational Biology* **7**(1-2), 95–114.

Kuang, R., Ie, E., Wang, K., Wang, K., Siddiqi, M., Freund, Y. & Leslie, C. (2005), 'Profile-based string kernels for remote homology detection and motif extraction', *Journal of Bioinformatics and Computational Biology* **3**(3), 527–550.

Lei, Z. & Dai, Y. (2005), 'An SVM-based system for predicting protein subnuclear localizations', *BMC Bioinformatics* **6**(1), 291.

Leslie, C., Eskin, E., Cohen, A., Weston, J. & Noble, W. (2004), 'Mismatch string kernels for discriminative protein classification', *Bioinformatics* **20**(4), 467–476.

Leslie, C., Eskin, E. & Grundy, W. S. (2002), The spectrum kernel: A string kernel for svm protein classification, *in* R. B. Altman, A. K. Dunker, L. Hunter, K. Lauerdale & T. E. Klein, eds, 'Proceedings of the Pacific Symposium on Biocomputing', World Scientific, pp. 564–575.

Leslie, C. & Kuang, R. (2004), 'Fast string kernels using inexact matching for protein sequences', *Journal of Machine Learning Research* **5**, 1435–1455.

Matthews, B. W. (1975), 'Comparison of predicted and observed secondary structure of t4 phage lysozyme', *Biochim Biophys Acta* **405**, 442–451.

Nakai, K. (2000), 'Protein sorting signals and prediction of subcellular localization', *Advances in Protein Chemistry* **54**, 277–344.

Neuberger, G., Maurer-Stroh, S., Eisenhaber, B., Hartig, A. & Eisenhaber, F. (2003), 'Motif refinement of the peroxisomal targeting signal 1 and evaluation of taxon-specific differences', *Journal of Molecular Biology* **328**(3), 567–579.

Park, K.-J. & Kanehisa, M. (2003), 'Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs', *Bioinformatics* **19**(13), 1656–1663.

Platt, J. (1999), Fast training of support vector machines using sequential minimal optimization, *in* B. Schölkopf, C. J. C. Burgess & A. J. Smola, eds, 'Advances in Kernel Methods–Suport Vector Learning', MIT Press, Cambridge, MA, pp. 185–208.

Rangwala, H. & Karypis, G. (2005), 'Profile-based direct kernels for remote homology detection and fold recognition', *Bioinformatics* **21**(23), 4239–4247.

Saigo, H., Vert, J.-P., Ueda, N. & Akutsu, T. (2004), 'Protein homology detection using string alignment kernels', *Bioinformatics* **20**(11), 1682–1689.

Schölkopf, B. & Smola, A. (2002), *Learning with kernels*, MIT Press, Cambridge, MA.

Schölkopf, B., Smola, A. & Müller, K.-R. (1999), Kernel principal component analysis, *in* B. Schölkopf, C. J. C. Burges & A. J. Smola, eds, 'Advances in Kernel Methods—Support Vector Learning', MIT Press, Cambridge, MA, pp. 327–352.

Wang, J., Sung, W.-K., Krishnan, A. & Li, K.-B. (2005), 'Protein subcellular localization prediction for gram-negative bacteria using amino acid subalphabets and a combination of multiple support vector machines', *BMC Bioinformatics* **6**(1), 174.

White, S. H. & von Heijne, G. (2005), 'Transmembrane helices before, during, and after insertion', *Current Opinion in Structural Biology* **15**(4), 378–386.

Higher order HMMs for Localization Prediction of Transmembrane Proteins

Stefan Maetschke **Mikael Bodén** **Marcus Gallagher**

School of Information Technology and Electrical Engineering,
The University of Queensland, Brisbane, Queensland 4072, Australia
Email: stefan@itee.uq.edu.au

Abstract

Utilizing the recently published LOCATE database, we construct Hidden Markov Models (HMMs) of first, second and third order for subcellular localization prediction of transmembrane proteins. In comparison with linear Support Vector Machines (SVMs), based on overall amino acid and di-peptide composition, higher order HMMs show a significant increase in prediction performance. The best performance was achieved by a second order HMM with a correlation coefficient of 0.46. A web-service for localization prediction of transmembrane proteins has been made available at http://pprowler.itee.uq.edu.au/TMPHMMLoc.

Keywords: HMM, SVM, subcellular localization, transmembrane protein

1 Introduction

Transmembrane proteins are inserted into the membranes of organelles and perform a variety of essential functions, such as channels, pumps, receptors and energy transducers. Current predictors for subcellular localization however, primarily target soluble proteins and ignore the characteristic topological domains of transmembrane proteins. On the other hand, topology predictors such as TMHMM (Sonnhammer, von Heijne & Krogh 1998, Krogh, Larsson, von Heijne & Sonnhammer 2001), Phobius (Käll, Krogh & Sonnhammer 2004) or HMMTOP (Tusnády & Simon 2001) are not designed for subcellular localization prediction.

Inspired by topology prediction methods, we construct a novel Hidden Markov Model (HMM) architecture for subcellular localization prediction of transmembrane proteins and compare it against two standard approaches for localization prediction of soluble proteins. More specifically, we 1) introduce the architecture and parameter estimation of the HMM, 2) measure the prediction accuracy and computation times of first, second and third order HMMs, 3) and compare the HMMs with linear Support Vector Machines (SVMs) that exploit overall amino acid and di-peptide composition as input. We utilize the recently published LOCATE database (Fink, Aturaliya, Davis, Zhang, Hanson, Teasdale, Kai, Kawai, Carninci, Hayashizaki & Teasdale 2006) and focus our comparison on five locations along the secretory pathway in mouse.

We thank Melissa Davis for many helpful discussions concerning transmembrane protein localization. This work was supported by the Australian Research Council Centre for Complex Systems.

2 Transmembrane proteins

Transmembrane proteins contain α-helical domains of hydrophobic residues that anchor the protein in the membrane. The *transmembrane domains* are usually flanked by *cap regions* that show a preference for charged residues and influence the orientation of the α-helix relative to the membrane (see Fig. 1). The more positively charged cap region of the transmembrane domain tends to reside on the cytosolic side (positive inside rule (von Heijne 1986)).

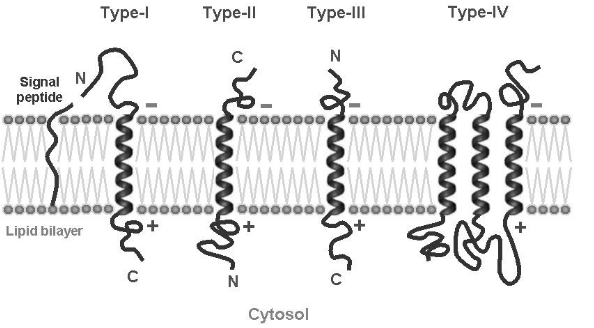


Figure 1: Transmembrane protein inserted into the lipid bilayer. The transmembrane domains form α-helices and the cap regions display a preference for charged residues (marked with plus and minus signs).

Four different types of transmembrane proteins can be distinguished[1] (Higy, Junne & Spiess 2004). Type-I proteins carry an N-terminal signal peptide which is cleaved when the protein is inserted into the membrane (von Heijne 1990). The N-terminus of the mature protein is at the lumenal or extracellular side and the C-terminus is at the cytoplasmic side. The orientation of Type-III is the same as Type-I proteins, whereas Type-II proteins are reversed. Multi-spanning proteins (Type-IV) pass the lipid bilayer several times with their termini on either side of the membrane (Rapoport, Goder, Heinrich & Matlack 2004).

Transmembrane proteins are localized to almost all compartments in the cell. We focus our study on organelles along the secretory pathway (see Fig. 2). The secretory pathway is especially complex due to its dynamic localization process that requires transmembrane proteins to travel through several stations until they reach their final destination (van Vliet, Thomas, Merino-Trigo, Teasdale & Gleeson 2003).

Entry station to the secretory pathway is the endoplasmic reticulum (ER). Transmembrane proteins are cotranslationally inserted into the ER membrane

[1] Note that this is a simple classification scheme that ignores important, but less frequent subtypes, such as reentrant regions in α-helical transmembrane proteins (Viklund, Granseth & Elofsson 2006).

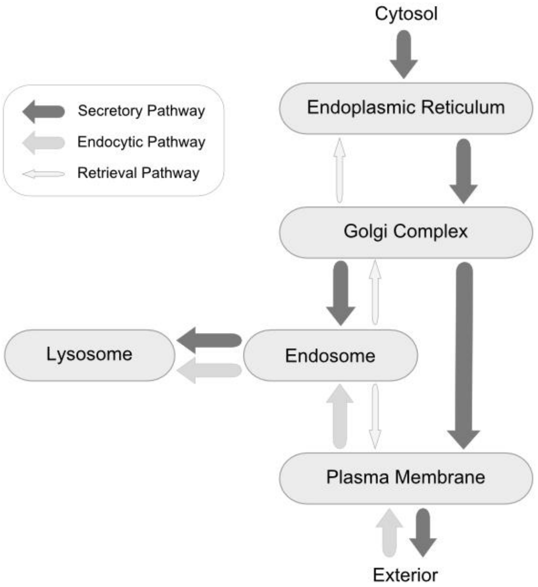


Figure 2: Schema of the secretory and endocytic pathways. The secretory pathway is directed from the interior of the cell to the exterior. The direction of the endocytic pathway is reversed.

and N-terminal signal peptides sequences are cleaved at this stage. Further transport occurs in vesicles that bud from the ER membrane and fuse with the Golgi complex (GO). At the Golgi complex, proteins are packed into coated vesicles and transported to the plasma membrane (PM) or the endosome (EN). From the endosome vesicles move proteins to the lysosome (LY). Also an indirect route exists, where proteins are exocytosed first and then internalized again, following the endocytic pathway. Additional retrieval pathways transport escaped proteins back to their original target location (van Vliet et al. 2003).

3 Related work

A multitude of prediction algorithms for protein subcellular localization have been developed. The vast majority of them is limited to soluble proteins however. We will discuss only a subset of the more recent algorithms that are related to our work.

Apart from methods that search for homologous or similarly annotated proteins in databases, the majority of current predictors exploit the amino acid or di-peptide composition and utilize SVMs to derive subcellular localization (Hua & Sun 2001, Park & Kanehisa 2003, Cui, Jiang, Liu & Ma 2004, Yu, Mendrola, Audhya, Singh, Keleti, DeWald, Murray, Emr & Lemmon 2004).

Composition based algorithms basically neglect the residue order of the sequence. To alleviate this weakness, autocorrelation functions (Feng & Zhang 2001), the pseudo amino acid composition (Chou 2001, Zhou & Doctor 2003) and the residue-coupling model (Guo, Lin & Sun 2005) have been applied.

A related approach is the partitioning of the protein sequence into sections (e.g. N-terminal, middle section, C-terminal) and the evaluation of section specific features such as amino acid composition and physicochemical properties (Small, Peeters, Legeal & Lurin 2004, Cui et al. 2004, Matsuda, Vert, Saigo, Ueda, Toh & Akutsu 2006). Yuan (1999) modeled the amino acid sequence directly with Markov chain models.

None of the aforementioned algorithms however, model the characteristic membrane spanning regions or consider the orientation of transmembrane proteins as topology predictors such as TMHMM (Krogh et al. 2001), Phobius (Käll et al. 2004) or HMMTOP (Tusnády & Simon 2001) do. The latter utilize detailed first order HMMs to describe the transmembrane, cap and loop regions but are not designed for subcellular localization prediction. The differences between topology prediction methods and our approach will be discussed in more detail in Section 6.

The only predictor for eukaryotic membrane proteins that we are aware of is based on amino acid composition and employs a least Mahalanobis distance classifier (Chou & Elrod 1999). A data set with 2105 membrane proteins extracted from Swiss-Prot (Release 35.0) with nine different locations was used and an overall jackknife accuracy of 65.9% was reported.

Since the data set was only weakly redundancy-reduced and contained different types of membrane proteins, these results are not comparable with ours. We compiled a strictly redundancy reduced, more recent data set, that contains transmembrane proteins only.

4 Data set

All predictors were trained and tested on protein data extracted from the LOCATE[2] database (Fink et al. 2006). LOCATE is based on the mouse transcriptome of the FANTOM3 Isoform Protein Sequence set (IPS7), enriched by membrane organization and subcellular localization annotation.

Membrane organization is determined by *MemO* (Davis, Zhang, Yuan & Teasdale 2006), a consensus method that employs SignalP (Bendtsen, Nielsen, von Heijne & Brunak 2004) and five transmembrane topology predictors (HMMTOP, TMHMM, SVMTM, MEMSAT, DAS) to predict signal peptides, transmembrane domains, protein orientation and subsequently protein type. Subcellular localization annotation in LOCATE is inferred from sources of varying quality (experimental, literature, predicted) but carefully reviewed.

We downloaded the XML version (`LOCATE_whole_db_v3-060810.xml`) of the database and extracted all transmembrane proteins with a unique subcellular localization annotation. The dataset was then filtered for proteins targeted to locations along the secretory pathway. Redundancy reduction was performed with BlastClust (Altschul, Gish, Miller, Myers & Lipman 1990), which removed all entries with a sequence similarity greater than 25%. The final data set contained 1351 transmembrane proteins with the following distribution: 873 plasma membrane (PM), 261 endoplasmic reticulum (ER), 141 Golgi apparatus (GO), 45 lysosome (LY), 31 endosome (EN).

5 Hidden Markov Models

A HMM is composed of a set of states $\{S_1, S_2, \ldots, S_N\}$ with transition probabilities a_{ij}. In the discrete case each state emits symbols v_k from a finite symbol set $V = \{v_1, v_2, \cdots, v_M\}$. The state transition probability distribution $\mathbf{A} = \{a_{ij}\}$ with $1 \leq i, j \leq N$, is defined as the probability that the model changes to state S_j given that it was in state S_i,

$$a_{ij} = P(q_{t+1} = S_j \mid q_t = S_i), \tag{1}$$

where q_t describes the state the model occupies at time step t. The symbol emission probabilities $\mathbf{B} =$

[2] `http://locate.imb.uq.edu.au`

$\{b_j(v_k)\}$ are the probabilities that symbol v_k is emitted (or observed) when the model is in state S_j with

$$b_j(v_k) = P(o_t = v_k \mid q_t = S_j), \qquad (2)$$

and o_t is the observed symbol at time step t (Durbin, Eddy, Krogh & Mitchison 1998).

Maximum likelihood estimates for state and emission probabilities can be directly calculated from labeled observation sequences or, for unlabeled data, gained in an unsupervised fashion utilizing a variant of the EM-algorithm (Baum-Welch). The most probable state sequence through the model is usually determined with a dynamic programming approach using the Viterbi-algorithm (Durbin et al. 1998).

In our domain, states describe sections of the protein sequence. For a first order HMM, V becomes the amino acid alphabet and t is a specific position within the sequence. Higher order HMMs are readily created by redefining V as an alphabet over pairs (second order) or n-tuples (n-th order) of amino acids (Durbin et al. 1998), and a protein is then processed as a sequence of overlapping, consecutive pairs or tuples of amino acids.

6 Localization predictor

The construction of the localization predictor can be divided into three phases. The first phase is the sequence labeling phase. The second phase is the construction of transmembrane protein models for each subcellular location based on the labeled sequences. In the third phase the protein models are aggregated in a localization model. In the following the three phases will be described in more detail.

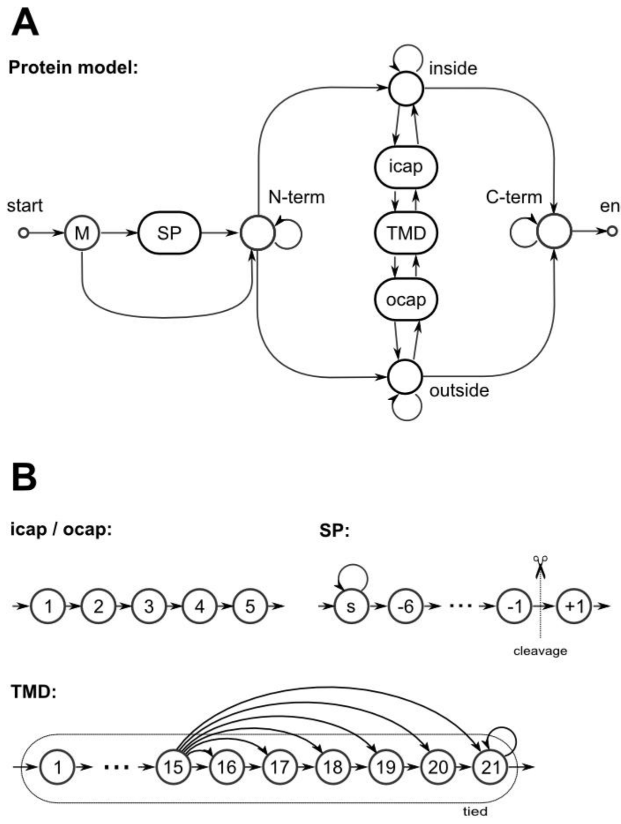


Figure 3: Prediction system. A) Transmembrane protein model for a single location. B) Details of the components of protein model A.

During the first phase every residue of the sequences in the training set is labeled with a state label of the protein model to construct (see Fig. 3). Labels are derived from existing sequence annotations such as transmembrane domains or signal peptides.

The first residue of the sequence is always labeled as Methionine (`M`). In the presence of a signal peptide annotation the following residues are labeled as signal peptide states. The position downstream of the annotated cleavage site is labeled `+1` and the adjacent six residues upstream are labeled `-1` to `-6`. The remaining upstream residues are all labeled as signal peptide residues `s` (See SP model in Fig. 3).

The ten amino acids following the signal peptide or the Methionine state are labeled as N-terminal (`N-term`). The transmembrane region is labeled with 15 up to 21 distinctive state labels (according to the length of the annotated region). The emission probabilities of these states are *tied* (each state uses the same emission probability distribution, See TMD model in Fig. 3). The five residues upstream and downstream of the transmembrane domain are labeled as inside (`icap`) or outside (`ocap`) regions, represented by five states (See icap/ocap model in Fig. 3).

The last ten residues of the sequence are labeled as C-Terminal (`C-term`) and all remaining amino acids are marked as `inside` or `outside` residues. The membrane orientation (N-terminus inside or outside) of the protein, which is required to label inside and outside residues and cap regions, is determined according to the presence or absence of an annotated signal peptide (a signal peptide indicates a non-cytosolic N-terminal). We also used the orientation annotation provided by the topology predictors in LOCATE but found it to result in lower prediction performance (data not shown). Likewise a fixed orientation (e.g. N-terminal always outside) was found to be inferior.

In phase two the labeled sequences are grouped according to the annotated subcellular localization. For each group a HMM is constructed. The model states are directly given by the used label set. Maximum likelihood estimates for emission and transition probabilities are derived from the frequencies of state residues and state transitions in the labeled sequences in the same way Profile-HMMs are built (Durbin et al. 1998). We also calculated the model parameters utilizing the unsupervised Baum-Welch algorithm (Durbin et al. 1998) but found the resulting prediction performance inferior to the supervised approach (data not shown).

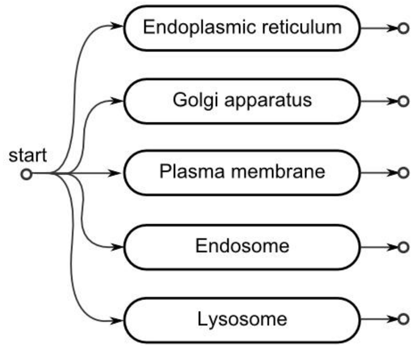


Figure 4: Aggregation of protein models within an overall HMM.

In phase three the transmembrane protein models, constructed in phase two, are aggregated in a single HMM with a unique start state but multiple end states (See part C of Fig. 4). Classification is performed by determining the Viterbi-path of the query sequence through the model and predicting the subcellular localization according to the end state of the most probable path.

Note that the protein models are smaller (less states and parameters) than similar models employed for topology prediction (Krogh et al. 2001, Käll et al. 2004, Tusnády & Simon 2001). There are three motivations for this: 1) The annotation of transmembrane regions and signal peptides in the training data is predicted, not experimentally confirmed. An overly refined model would only predict predicted data with high accuracy. 2) The objective is to predict subcellular localization, not topology. The exact domain borders are therefore of secondary interest. 3) The data sets for some locations (e.g. endosome, lysosome) are very small and parameters for more complex models cannot be estimated reliably.

7 Results

Many algorithms for subcellular localization prediction of *soluble proteins* are based on SVMs that exploit the overall amino acid or di-peptide composition of a protein as input (Hua & Sun 2001, Park & Kanehisa 2003). We therefore compare the prediction accuracy, and training and query time, for two composition based SVMs with HMMs of varying order. In the following, SVM1 denotes a linear SVM that exploits the amino acid composition and SVM2 is a linear SVM that utilizes the di-peptide composition. HMM1, HMM2 and HMM3 refer to first, second and third order HMMs, respectively.[3]

The results in Table 1 show a significant increase in prediction accuracy of higher order HMMs compared to first order HMMs or SVMs. Notably the correlation coefficient of SVM1 is a magnitude smaller than that of SVM2. This suggests that the dipeptides composition is a much better representation of the typical sorting signals (e.g. ER retrieval signal `K(X)KXX` or lysosomal/endosomal di-leucine targeting signal) than the mono amino acid composition.

Concerning training and query time, the HMMs are fast to train but slow to query while the situation for the SVMs is reversed. The training time for the third order HMM is surprisingly high. We believe that the physical memory (1GB) was not sufficient and memory swapping took place in this case.

Classes with small numbers of training samples, such as the endosomal (EN) and lysosomal (LY) classes, cause a clear drop in prediction performance for higher order HMMs. SVM2, as a maximum margin classifier, is less effected by this difficulty, while the performance of SVM1 is poor in general. There is no significant difference in prediction performance between second and third order HMMs but the second order model features lower query times and memory requirements.

PM	ER	GO	EN	LY	
834	25	11	3	0	**PM**
125	**126**	8	1	1	**ER**
63	22	**54**	0	2	**GO**
21	0	1	**9**	0	**EN**
28	4	1	0	**12**	**LY**

Table 2: Ten-fold cross-validation confusion matrix for second order model (HMM2). Rows represent observed locations and columns represent predicted locations.

To gain a deeper insight into the prediction performance of the second order HMM2, we calculated the ten-fold cross-validation confusion matrix (see Table 2). The confusion matrix shows that most of the misclassified proteins are predicted as targeted to the plasma membrane (left most column). This is not surprising, since the plasma membrane class is the majority class. Also the plasma membrane is known to serve as a default location for proteins that lack specific sorting signals (Pedrazzini, Villa & Borgese 1996, Brandizzi, Frangne, Marc-Martin, Hawes, Neuhaus & Paris 2002). Interestingly, there is no confusion between endosomal and lysosomal targeted proteins and in general little confusion between proteins targeted to non-plasma membrane locations. This indicates that the current location models seem to miss some specific targeting signal, and that more sensitive models can increase the prediction accuracy without severing the discrimination between locations.

8 Conclusion

We presented a novel architecture of an HMM based localization predictor for transmembrane proteins. In contrast to topology predictors, the new architecture has less states but models the terminal regions and is of second order. The latter is in agreement with the observation that location predictors based on dipeptide composition typically achieve higher performance than classifiers that exploit the mono amino acid composition only.

By modeling the characteristic topology of transmembrane proteins, the new predictor achieves a significant increase in prediction accuracy (correlation coefficient 0.46), compared to predictors based on overall di-peptide composition. To our knowledge, it is the only localization predictor specifically for transmembrane proteins, that is currently available online (`http://pprowler.itee.uq.edu.au/TMPHMMLoc`).

We took advantage of the recently published LOCATE database and concentrated our efforts on locations along the secretory pathway, which are especially difficult to distinguish between.

Further work will focus on extending the range of predicted locations, utilizing additional data sources and comparing the new predictor against a more comprehensive set of alternative methods.

References

Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. (1990), 'Basic local alignment search tool.', *J Mol Biol* **215**(3), 403–410. *http://dx.doi.org/10.1006/jmbi.1990.9999

Bendtsen, J. D., Nielsen, H., von Heijne, G. & Brunak, S. (2004), 'Improved prediction of signal peptides: SignalP 3.0.', *J Mol Biol* **340**(4), 783–795. *http://dx.doi.org/10.1016/j.jmb.2004.05.028

Brandizzi, F., Frangne, N., Marc-Martin, S., Hawes, C., Neuhaus, J. & Paris, N. (2002), 'The destination for single-pass membrane proteins is influenced markedly by the length of the hydrophobic domain', *Plant Cel* **14**, 1077–1092.

Chou, K. C. (2001), 'Prediction of protein cellular attributes using pseudo-amino acid composition.', *Proteins* **43**(3), 246–255.

Chou, K. C. & Elrod, D. W. (1999), 'Prediction of membrane protein types and subcellular locations', *Proteins* **35**, 137–153.

[3] Note that second and third order HMMs utilize the same architecture as described above but observe amino acid pairs or triples instead of single amino acids.

Method	ER	GO	PM	EN	LY	Overall	TT	QT
SVM1	0.072	0.000	0.049	0.000	0.000	**0.024** (± 0.007)	0.003	0.001
SVM2	0.318	0.304	0.320	0.369	0.235	**0.309** (± 0.017)	0.347	0.001
HMM1	0.172	0.232	0.240	0.160	0.034	**0.168** (± 0.012)	0.028	0.660
HMM2	0.492	0.479	0.506	0.403	0.433	**0.462** (± 0.020)	0.025	0.782
HMM3	0.532	0.455	0.469	0.385	0.392	**0.447** (± 0.019)	0.157	0.934

Table 1: Prediction accuracy and training and query times for methods split by location. Results are 10 fold cross-validated, 10 times repeated. Method = prediction method, ER = Endoplasmic Reticulum, GO = Golgi Complex, PM = Plasma Membrane, EN = Endosome, LY = Lysosome. Overall = overall mean correlation coefficient with 95% confidence interval in brackets. A correlation coefficient of 1.0 is ideal. TT = training time in msec and QT = query time in msec per sample on a Pentium 4, 2.8 GHz with 1 GB main memory.

Cui, Q., Jiang, T., Liu, B. & Ma, S. (2004), 'Esub8: A novel tool to predict subcellular localizations in eukaryotic organisms', *BMC Bioinformatics* **5**(66).

Davis, M. J., Zhang, F., Yuan, Z. & Teasdale, R. D. (2006), 'MemO: A consensus approach to the annotation of a protein's membrane organization', *In Silico Biol.* **6**(0037).

Durbin, R. M., Eddy, S. R., Krogh, A. & Mitchison, G. (1998), *Biological sequence analysis*, Cambridge University Press, Cambridge, UK.

Feng, Z. P. & Zhang, C. T. (2001), 'Prediction of the subcellular location of prokaryotic proteins based on the hydrophobicity index of amino acids.', *Int. J. Biol. Macromol.* **28**(3), 255–261.

Fink, L., Aturaliya, R., Davis, M., Zhang, F., Hanson, K., Teasdale, M., Kai, C., Kawai, J., Carninci, P., Hayashizaki, Y. & Teasdale, R. (2006), 'LOCATE: a mouse protein subcellular localization database.', *Nucleic Acids Research* **34 (Database issue)**, D213–D217. *http://dx.doi.org/10.1093/nar/gkj069

Guo, J., Lin, Y. & Sun, Z. (2005), A novel method for protein subcellular localization: Combining residue-couple model and SVM, *in* Y.-P. P. Chen & L. Wong, eds, 'Proceedings of 3rd Asia-Pacific Bioinformatics Conference', Imperial College Press.

Higy, M., Junne, T. & Spiess, M. (2004), 'Topogenesis of membrane proteins at the endplasmic reticulum', *Biochemistry* **43**, 12716–12722.

Hua, S. & Sun, Z. (2001), 'Support vector machine approach for protein subcellular localization prediction', *Bioinformatics* **17**(8), 721–728.

Käll, L., Krogh, A. & Sonnhammer, E. (2004), 'A combined transmembrane topology and signal peptide prediction method', *Journal of Molecular Biology* **338**(5), 1027–1036.

Krogh, A., Larsson, B., von Heijne, G. & Sonnhammer, E. L. (2001), 'Predicting transmembrane protein topology with a hidden Markov model: Application to complete genomes.', *J Mol Biol* **305**(3), 567–580. *http://dx.doi.org/10.1006/jmbi.2000.4315

Matsuda, S., Vert, J.-P., Saigo, H., Ueda, N., Toh, H. & Akutsu, T. (2006), 'A novel representation of protein subsequences for prediction of subcellular location using support vector machines', *Protein Science* **14**, 2804–2813.

Park, K.-J. & Kanehisa, M. (2003), 'Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs', *Bioinformatics* **19**(13), 1656–1663.

Pedrazzini, E., Villa, A. & Borgese, N. (1996), 'A mutant cytochrome b(5) with a lengthened membrane anchor escapes from the endoplasmic reticulum and reaches the plasma membrane', *Proc. Natl. Acad. Sci. USA* **93**, 4207–4212.

Rapoport, T. A., Goder, V., Heinrich, S. U. & Matlack, K. E. S. (2004), 'Membrane-protein integration and the role of the translocation channel.', *Trends Cell Biol* **14**(10), 568–575. *http://dx.doi.org/10.1016/j.tcb.2004.09.002

Small, I., Peeters, N., Legeal, F. & Lurin, C. (2004), 'Predotar: A tool for rapidly screening proteomes for N-terminal targeting sequences', *Proteomics* **4**, 1581–1590.

Sonnhammer, E. L., von Heijne, G. & Krogh, A. (1998), 'A hidden Markov model for predicting transmembrane helices in protein sequences.', *Proc Int Conf Intell Syst Mol Biol* **6**, 175–182.

Tusnády, G. E. & Simon, I. (2001), 'The HMMTOP transmembrane topology prediction server', *Bioinformatics* **17**(9), 849–850.

van Vliet, C., Thomas, E., Merino-Trigo, A., Teasdale, R. & Gleeson, P. (2003), 'Intracellular sorting and transport of proteins', *Progress in Biophysics & Molecular Biology* **83**, 1–45.

Viklund, H., Granseth, E. & Elofsson, A. (2006), 'Structural classification and prediction of reentrant regions in α-helical transmembrane proteins: Application to complete genomes.', *J Mol Biol* **doi:10.1016/j.jmn.2006.06.037**.

von Heijne, G. (1986), 'The distribution of positively charged residues in bacterial inner membranes correlates with the trans-membrane topology.', *EMBO J* **5**, 3021–3027.

von Heijne, G. (1990), 'The signal peptide', *Journal of Membrane Biology* **115**, 195–201.

Yu, J. W., Mendrola, J. M., Audhya, A., Singh, S., Keleti, D., DeWald, D. B., Murray, D., Emr, S. D. & Lemmon, M. A. (2004), 'Genome-wide analysis of membrane targeting by *S. cerevisiae* pleckstrin homology domains.', *Molecular Biology of the Cell* **13**(5), 677–688.

Yuan, Z. (1999), 'Prediction of protein subcellular locations using Markov chain models.', *FEBS Lett.* **451**(1), 23–26.

Zhou, G.-P. & Doctor, K. (2003), 'Subcellular location prediction of apoptosis proteins.', *Proteins* **50**(1), 44–48. *http://dx.doi.org/10.1002/prot.10251

Multi-stage Redundancy Reduction: Effective Utilisation of Small Protein Data Sets

John Hawkins **Mikael Bodén**

School of Information Technology and Electrical Engineering
The University of Queensland,
St Lucia, QLD, Australia,
Email: jhawkins@itee.uq.edu.au

Abstract

In many important bioinformatics problems the data sets contain considerable redundancy due to the evolutionary processes which generate the data and biases in the data collection procedures. The standard practice in bioinformatics involves removing the redundancy such that there is no more than at most forty percent similarity between sequences in a data set. For small data sets this can dilute the already impoverished data beyond the boundary of practicality. One can choose to include all available data in the process by just ensuring that only the training and test samples have the required redundancy gap. However, this encourages overfitting of the model by exposure to a highly redundant training sets. We outline a process of multi-stage redundancy reduction, whereby the paucity of data can be effectively utilised without compromising the integrity of the model or the testing procedure.

Keywords: Redundancy Reduction, Generalisation Estimation, Cross Validation

1 Introduction

An essential part of protein data set development for machine learning applications in bioinformatics involves removing redundancy so that the bias in the data is minimised (Hobohm, Scharf, Schneider & Sander 1992). The redundancy reduction helps prevent a model over fitting to the bias in the data collection processes, and prevents predictive accuracy being overestimated.

However, there are a number of biological problems where the data sets are comparatively small simply due to the fact that they relate to subtle aspects of cellular life. In these instances we are faced with a problem: "How to train and test our models so that we best utilise the available data and do not bias our tests?".

One solution that has been proposed to this problem involves giving a weighting to each of the data points to correct for biases in the training data (Krogh & Mitchison 1995, Eddy, Mitchison & Durbin 1995). The prime difficulty posed by such an approach is that not all learning algorithms are conducive to using weighted samples. In order to employ this technique one needs to restrict the type of model used, or modify an existing model to accommodate the weightings. Recently the weighting of samples has been extended to the calculation of performance metrics (Budagyan & Abagyan 2006). The authors concluded that one need not perform an artificial reduction of the data set if the testing is performed using a weighted metric. Their results indicated that inclusion of redundant data can improve the performance of the model.

We present a simple solution to the problem of making effective use of small data sets without compromising testing rigour. The technique takes the form of a regime for gradual data set reduction as the model moves from training to testing. The redundancy reduction occurs in three stages, allowing small amounts of redundancy within the training sets, but rigorously excluding it between training and test sets. Furthermore, we provide a clear indication of generalisation improvement offered by redundancy reduction by showing that performance is optimal when not using all available data, but by allowing only small amounts of redundancy within the training sets.

2 Background

The effective application of machine learning techniques to problems of classification is not simply a matter of training a model on all available data. Although a model produced in this fashion will perform very well on data with similarity to the training data it will tend to fail on genuinely novel data. Hence, the performance statistics produced by cross-validation on data sets containing redundancy will not be a reliable guide to their ability to generalise.

The problem stems from two sources, firstly used a data set that over represents some region of the problem space encourages the model to over fit this data. This is less of a problem if the bias is present in the real world, however often these biases are due to the collection of data. The second problem with redundant data is that when used to estimate the performance of the model it will bias those estimates due to the fact that the test points are very close to repetitions of the training points. When a model that is trained and tested on redundant data "the apparent predictive performance may be overestimated, reflecting the method's ability to reproduce its own particular input rather than its generalization power" (Baldi & Brunak 2001) (page 6).

Considerable effort has gone into developing algorithms to minimise the redundancy within the data yet maximise the amount left with which to build models (Hobohm et al. 1992). A *de facto* standard has emerged in bioinformatics to perform a redundancy reduction of data sets using sequence homology scores, such that no two sequences have greater than 25%-40% identical residues across a specified length. This threshold is no doubt due to the fact that it is the so called 'twilight' region in which sequence alignments become a poor indicator of homology (Rost 1999).

However, what remains unaddressed is whether the two reasons for redundancy reduction require the same level of reduction in order to mitigate their respective problems. This issue forms the central question of this study and the answer to which suggests the technique of multi-stage redundancy reduction as a method of effectively utilising small data sets.

3 The Method

The essence of the technique is as follows: we perform an initial redundancy reduction of the data in order to remove the sequences that are most similar. The redundancy reduction is performed using BLASTCLUST to generate clusters with a specified level of similarity. From these clusters a single sample is chosen as the representative of the cluster. The *initial reduction threshold*, Θ_i, is a permissive threshold, between 40% and 90% similarity. The sequences remaining after the initial reduction comprise the data set for the purposes of training the model.

In order to ensure that the testing procedure is not compromised by the existence of some redundancy in the data, we perform a second clustering at the *final reduction threshold*, Θ_f, set to a low enough level as to guarantee a rigorous testing. These clusters are then used to generate the subsets of data for the cross-validations. Each of the clusters is allocated to one of the subsets. So that all the redundancy exists within the cross validation subsets. This ensures that there is no redundancy between the sequences that are used for training and those used for testing, such that the cross-validation is assured to be an adequate test of generalisation.

The cross-validation is then performed such that the partially redundant data is used in the training of the models. However, in each iteration, the set that has been allocated for testing undergoes a further reduction at the *final reduction threshold* Θ_f, to remove the remaining redundancy. In this way we allow the models to utilise some redundancy in the training data, but remove all redundancy from the testing procedure.

The procedure is demonstrated schematically in Figure 1.

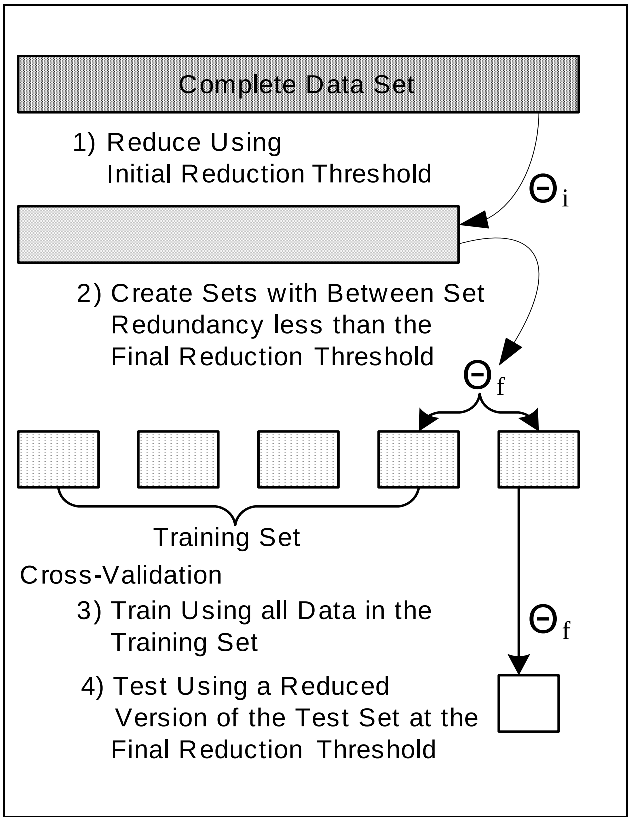

Complete Data Set
1) Reduce Using Initial Reduction Threshold
Θ_i
2) Create Sets with Between Set Redundancy less than the Final Reduction Threshold
Θ_f
Training Set
Cross-Validation
3) Train Using all Data in the Training Set
Θ_f
4) Test Using a Reduced Version of the Test Set at the Final Reduction Threshold

Figure 1: Schematic Representation of the Multi-Stage Redundancy Reduction Training and Testing Procedure.

4 Data sets

For the purposes of demonstrating the utility of the technique we apply it to two different data sets of proteins. Both data sets revolve around the importation of proteins into the peroxisome, which is a small but important organelle in eukaryotic cells. The two mechanisms are named after the sequence signals which the proteins rely on for recognition and import. Peroxisomal Targeting Signal One (PTS1) and Peroxisomal Targeting Signal Two (PTS2), are both subtle distinct sequences within the protein that allow import into the organelle through distinct protein pathways (Baker & Sparkes 2005, Michels, Moyersoen, Krazy, Galland, Herman & Hannaert 2005). The crucial aspect of these data sets for the current paper is that the peroxisome has a small protein complement, hence the data sets are small.

The PTS1 pathway relies on an C-terminal tripeptide and sequence of nine preceding residues that support its recognition. We have outlined the extraction of this data set in previous studies (Wakabayashi, Hawkins, Maetschke & Bodén 2005, Hawkins & Bodén 2005). The data set derivation relies on a biologically informed template for the import signal. This template fits all known positives, but due to its generality fits a larger number of negatives.

The PTS2 pathway relies on a 9-mer with an unspecified position, although most instances occur in the N-terminal region. There are far fewer known instances of PTS2 proteins, and to make matters worse the template signal is less specific. Hence, the training sets for differentiating between real and fallacious PTS2 instances are highly unbalanced. We have outlined the extraction of this data set in a previous study (Bodén & Hawkins 2006). Due to the size of the negative set we maintain the heavily reduced version (2799 proteins reduced at a 10% threshold) and instead focus the multi-stage redundancy reduction on the positive set only.

The numbers of proteins that result from the varying levels of redundancy reduction are shown in Table 1.

5 Simulations

In order to demonstrate the effectiveness of the technique and identify a threshold for the initial redundancy reduction we perform a range of simulations over each of the data sets. For each data set we produce a set of versions each of which have been through an initial redundancy reduction. For these initial reductions we use a set of thresholds varying from 100 (for no reduction) down to 30. The number of proteins in the data sets for each of these threshold is shown in Table 1.

As a model to train on these problems we have chosen a Support Vector Machine with a spectrum kernel. The spectrum kernel is a general purpose sequence kernel that is efficient to run and has proven effective on a wide range of bioinformatics problems. The spectrum kernel takes one parameter, k which defines the length of the sequence segments considered in the spectrum.

For a given sequence, the spectrum of the sequence

		Redundancy Reduction Threshold									
Data set		100	95	90	85	80	70	60	50	40	30
PTS1	Positives	139	131	114	97	83	66	62	56	50	47
	Negatives	291	256	230	208	193	180	171	165	164	163
PTS2	Positives	97	91	81	69	62	56	51	41	37	35

Table 1: Numbers of Proteins in each data set as the initial redundancy reduction threshold is varied. At the upper limit, a threshold of 100 means no reduction is performed. At the lower end of 30% the threshold is identical to that used to distinguish the test sets, hence the process is equivalent to the standard process of a single redundancy reduction prior to training and testing.

is the set of all k-mers it contains. The Spectrum kernel compares any two sequences by considering the number of these k-mers that two sequences share (Leslie, Eskin & Grundy 2002). More specifically, the kernel calculates the dot product between the vectors holding all k-mer counts for any pair of sequences. If two sequences share a large number of k-mers they produce a large spectrum kernel value.

For both data sets we explore k values ranging between 1 and 5. We run these on each of the data sets, such that the initial reduction threshold ranges from 100% (No Reduction) to 30% (Single Stage Reduction). We run each configuration as a ten-fold cross-validation, ten times using a different seed to split the data for the cross validations.

We use our own implementation of the spectrum kernel that runs with a modified version of the LIBSVM package (Chang & Lin 2001). The C value of an SVM is commonly called the regularisation constant and indicates the penalty that is applied to samples that are positioned on the wrong side of the decision boundary. For the PTS1 problem we use a general C value of 0.5, however for the PTS2 problem, due to the massive imbalance of the data, we modified the code such that the positive samples us $C = 1000$ and the negative samples use $C = 0.002$. These values were found through a number of trial runs as a method of forcing the learning to give equal emphasis to both classes.

6 Results

The results for each of the spectrum kernels on the PTS1 problem are shown in Table 2. In each case we shown the mean Matthews' Correlation Coefficient MCC over the ten independent runs. The MCC is calculated using the formula:

$$r(c) = \frac{tp_c tn_c - fp_c fn_c}{\sqrt{(tp_c + fn_c)(tp_c + fp_c)(tn_c + fp_c)(tn_c + fn_c)}}. \quad (1)$$

The data are shown graphically in Figure 2 with a standard error bar showing the estimated standard deviation of the mean. This is calculated with the formula:

$$stderr = \frac{\sigma}{\sqrt{N}}. \quad (2)$$

Where σ is the standard deviation of the test statistic, in this case the MCC, and N is the number of samples, in this case 10.

For three of the four k values tried the multi-stage redundancy reduction technique produced the best model. The best overall model produced used an initial redundancy reduction threshold of 85% and a k value of 2. As we can see in Figure 2, the results are somewhat variable across the different kernels and thresholds. In spite of the lack of a consistent trend, if we compare the results of the best model at threshold 30 with the best overall model, the standard error bar indicates that using some redundancy in the training produces significantly better results than not. However, due to the overlap of the standard error distributions it is not possible to say whether the apparent improvement offered by the multi-stage redundancy reduction is significant.

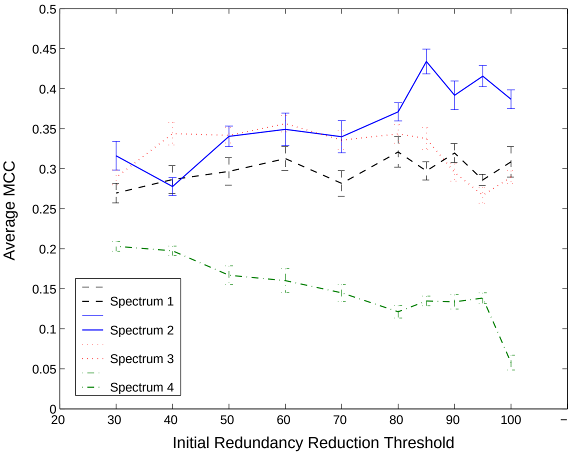


Figure 2: The average MCC for the Spectrum Kernel on the PTS1 problem, plotted against the initial threshold of redundancy reduction. The error bars depict one standard error on either side of the mean. Data generated from ten runs of ten-fold cross-validation.

Similarly the results for each of the spectrum kernels on the PTS2 problem are shown in Table 3.

For the PTS2 problem we see that for all four of the spectrum values used the best performing kernel was produced under multi-stage redundancy reduction. The best overall model produced used an initial redundancy reduction threshold of 85% and a k value of 4. As we can see in Figure 3, the results are much more consistent across the different kernel settings. Three of the four performing best with an initial threshold between $80-85\%$. The effect is most pronounced with $k = 4$, where under the standard practice or using all data the kernel performs very poorly. In this case the best model performs significantly better under the multi-stage redundancy reduction scheme than either using all data, or single-stage redundancy reduction.

It is interesting to note that in both case studies the majority of the kernels perform best when allowed to use some of the redundant data. In some cases profoundly better than if all that data was used or it was simply discarded. What we see clearly in both figures is that the best model for each problem was created using the mutli-stage redundancy reduction procedure with an initial reduction of 85%.

PTS1 Simulation Results

	Initial Redundancy Reduction Threshold									
k-value	100	95	90	85	80	70	60	50	40	30
1	0.3087	0.2861	0.3197	0.2973	0.3210	0.2817	0.3127	0.2967	0.2866	0.2697
2	0.3868	0.4158	0.3918	0.4341	0.3711	0.3401	0.3492	0.3405	0.2778	0.3163
3	0.2896	0.2669	0.2954	0.3376	0.3436	0.3357	0.3563	0.3418	0.3439	0.2892
4	0.0578	0.1384	0.1337	0.1348	0.1213	0.1449	0.1602	0.1669	0.1974	0.2031

Table 2: Average MCC values for the Spectrum Kernel run with different k-values and different thresholds for the initial redundancy reduction. The first column with an initial threshold of 100% involves using all available data to train the models. The final column, at 30%, involves a single redundancy reduction prior to training and testing. Intermediate values are the result of the two stage redundancy reduction procedure.

PTS2 Simulation Results

	Initial Redundancy Reduction Threshold									
k-value	100	95	90	85	80	70	60	50	40	30
1	0.0823	0.0812	0.0832	0.0873	0.0842	0.0846	0.0902	0.0868	0.0973	0.0903
2	0.0886	0.0872	0.1107	0.1291	0.1281	0.1045	0.0858	0.0812	0.0533	0.0507
3	0.0797	0.0818	0.1212	0.1369	0.1444	0.0648	0.0336	0.0205	0.0116	0.0356
4	0.0271	0.0292	0.0295	0.1588	0.1419	-0.0035	-0.0040	-0.0031	-0.0030	-0.0029

Table 3: Average MCC values for the Spectrum Kernel run with different k-values and different thresholds for the initial redundancy reduction. The first column with an initial threshold of 100% involves using all available data to train the models. The final column, at 30%, involves a single redundancy reduction prior to training and testing. Intermediate values are the result of the two stage redundancy reduction procedure.

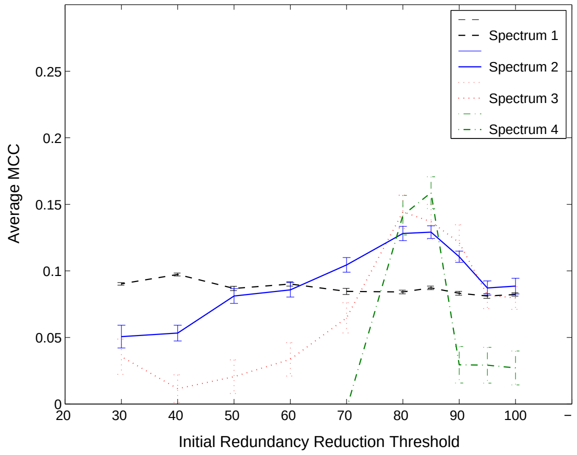


Figure 3: The average MCC for the Spectrum Kernel on the PTS2 problem, plotted against the initial threshold of redundancy reduction. The error bars depict one standard error on either side of the mean. Data generated from ten runs of ten-fold cross-validation.

7 Conclusion

Some key biological problems have only small amounts of data available for the building of models. It is therefore crucial to make effective use of the paucity of available data. Data sets typically under go a process of redundancy reduction for two reasons: To prevent the model from over fitting to the bias present in the data, and to ensure that our testing of the model's generalisation ability is rigorous. Typically the redundancy reduction is done once as the data set is curated with the implicit assumption that the threshold for reduction should be identical for both of these purposes.

We have shown that by treating these two purposes of redundancy reduction separately, we are able to increase the amount of data available to train our models. By using rigorous testing procedures we have shown that the optimal thresholds of redundancy for these two purposes are not identical. I.e the reduction required in order to prevent over fitting is less than that required to perform rigorous testing. It appears that one can produce a superior model by allowing it to train on data with a mild amount of redundancy.

References

Baker, A. & Sparkes, I. A. (2005), 'Peroxisome protein import: some answers, more questions', *Current Opinion in Plant Biology* **8**(6), 640–647.

Baldi, P. & Brunak, S. (2001), *Bioinformatics : the machine learning approach, Second Edition*, MIT Press, Cambridge, Mass.

Bodén, M. & Hawkins, J. (2006), Evolving discriminative motifs for recognizing proteins imported to the peroxisome via the pts2 pathway, *in* 'Proceedings of the IEEE Congress on Evolutionary Computation', IEEE, Canada, pp. 9300–9305.

Budagyan, L. & Abagyan, R. (2006), 'Weighted quality estimates in machine learning', *Bioinformatics* p. btl458.

Chang, C.-C. & Lin, C.-J. (2001), *LIBSVM: a library for support vector machines*. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

Eddy, S. R., Mitchison, G. J. & Durbin, R. (1995), 'Maximum discrimination hidden markov models of sequence consensus.', *Journal of Computational Biology* **2**(1), 9–23.

Hawkins, J. & Bodén, M. (2005), Predicting peroxisomal proteins, *in* 'Proceedings of the IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology', IEEE, Piscataway, pp. 469–474.

Hobohm, U., Scharf, M., Schneider, R. & Sander, C. (1992), 'Selection of representative protein data sets', *Protein Science* **1**(3), 409–417.

Krogh, A. & Mitchison, G. (1995), Maximum entropy weighting of aligned sequences of proteins or dna, *in* 'Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology', pp. 215–221.

Leslie, C., Eskin, E. & Grundy, W. S. (2002), The spectrum kernel: A string kernel for svm protein classification, *in* R. B. Altman, A. K. Dunker, L. Hunter, K. Lauerdale & T. E. Klein, eds, 'Proceedings of the Pacific Symposium on Biocomputing', World Scientific, pp. 564–575.

Michels, P., Moyersoen, J., Krazy, H., Galland, N., Herman, M. & Hannaert, V. (2005), 'Peroxisomes, glyoxysomes and glycosomes', *Molecular Membrane Biology* **22**(1 - 2), 133–145.

Rost, B. (1999), 'Twilight zone of protein sequence alignments', *Protein Eng.* **12**(2), 85–94.

Wakabayashi, M., Hawkins, J., Maetschke, S. & Bodén, M. (2005), Exploiting targetting signal dependencies in the prediction of pts1 peroxisomal proteins, *in* M. Gallagher, J. Hogan & F. Maire, eds, 'Intelligent Data Engineering and Automated Learning - IDEAL 2005: 6th International Conference', Vol. 3578 of *Lecture Notes in Computer Science*, Springer, pp. 454–461.

Linear Predictive Coding and its Decision Logic for Early Prediction of Major Adverse Cardiac Events using Mass Spectrometry Data

Tuan D. Pham[1,2,†], **Honghui Wang**[3], **Xiaobo Zhou**[4], **Dominik Beck**[1], **Miriam Brandl**[1], **Gerard Hoehn**[3], **Joseph Azok**[3], **Marie-Luise Brennan**[5], **Stanley L. Hazen**[5], **King Li**[5], **and Stephen T.C. Wong**[4]

[1]Bioinformatics Applications Research Center
[2]School of Mathematics, Physics, and Information Technology
James Cook University
Townsville, QLD 4811, Australia
[3]Clinical Center, National Institutes of Health
Bethesda, MD 20892, USA
[4]HCNR Center for Bioinformatics, Harvard Medical School
Boston, MA 02115, USA
[5]Center for Cardiovascular Diagnostics and Prevention, Cleveland Clinic Foundation
Cleveland, OH 44195, USA
[†]Email: tuan.pham@jcu.edu.au

Abstract

Proteomics is an emerging field of modern biotechnology and an attractive research area in bioinformatics. Protein annotation by mass spectrometry has recently been utilized for the classification and prediction of diseases. In this paper we apply the theory of linear predictive coding and its decision logic for the prediction of major adverse cardiac risk using mass spectra. The new method was tested with a small set of mass spectrometry data. The initial experimental results are found promising for the prediction and show the implication of the potential use of the data for biomarker discovery.

Keywords: Proteomics, mass spectrometry, major adverse cardiac events, classification, prediction, theory of linear prediction.

1 Introduction

Besides genomics, life-science researchers study proteomics in order to gain insight into the functions of cells by learning how proteins are expressed, processed, recycled, and their localization in cells. Proteomics is simply the study of proteome which refers to the entire set of expressed protein in a cell. Proteomics can be divided into two categories: expression proteomics and cell-map proteomics (Weir et al. 2003).

Protein expression profiles or expression proteomics can be used for large-scale protein characterization or differential expression analysis that has many applications such as biomarker discovery for disease classification and prediction, new drug treatment and development, virulence factors, and polymorphisms for genetic mapping, and species determinants (Griffin et al. 2001, Aebersold & Mann 2003, Weir et al. 2003). In comparison with transcriptional profiling in functional genomics, proteomics has some obvious advantages in that it provides a more direct approach to studying cellular functions because most gene functions are characterized by proteins (Xiong 2006). Cell-map proteomics is large-scale characterization of protein interactions and an integrated view of cellular processes at the protein level.

The identities of expressed proteins in a protemome can be determined by protein separation, identification, and quantification. One of many approaches for separating proteins involves two-dimensional gel electrophoresis followed by gel image processing. Once proteins are separated, protein differential expression can be characterized using mass spectrometry (MS), which is a high-resolution technique for determining molecular masses and provides rapid and accurate measurement of protein profiling in complex biological and chemical mixture. Protein profiling of plasma and serum can be prepared with a matrix-assisted laser desorption ionization (MALDI) ion source or the surface-enhanced laser desorption ionization (SELDI) ion source coupled to a time-of-flight (TOF) mass analyzer with a chevron micochannel plate detector. Detailed discription on mass spectrometry and its advanced developments can be found in the review by Shin and Markey (2006).

Proteomic patterns have recently been used for early detection of cancer progressions (Sauter et al. 2002, Petricoin et al. 2002, Conrads et al. 2003). Obviously, early detection of such diseases has the potential to reduce mortality. In fact, it has been foreseen that advances in mass-spectrometry based diagnostics may lead to a new revolution in the field of molecular medicine (Petricoin & Liotta 2003, Conrads et al. 2003, Wulfkuhle et al. 2003).

Methods for classification of normal and cancerous states using mass spectrometry data have been recently developed. Petricoin *et al.* (2002) applied cluster analysis and genetic algorithms to detect early stage ovarian cancer using proteomic spectra. Lilien *et al.* (2003) applied principal component analysis and a linear discriminant function to classify ovarian and prostate cancers. Sorace and Zhan (2003) used mass spectrometry serum profiles to detect early ovarian cancer. Wu *et al.* (2003) compared the performance of several methods for the classification of mass spectrometry data. Tibshirani *et al.* (2004) proposed a probabilistic approach for sample classification from protein mass spectrometry data. Morris *et al.* (2005) applied wavelet transforms and peak detection for feature extraction of MS data. Yu *et al.* (2005) developed a method for dimensionality reduction for high-throughput MS data. Levner (2005) used feature selection methods and then applied the nearest centroid technique to classify MS-based ovarian and prostate cancer datasets. Given the promising integration of machine-learning methods and mass spectrometry data in high-throughput proteomics (Shin & Markey 2006), this new biotechnology still encounters several challenges in order to become a mature platform for clinical diagnostics and protein-based biomarker profiling. Some of major challenges include noise filtering of MS data, selection of computational methods for MS-based classification, feature extraction and feature reduction of MS datasets.

The motivation of this research has been initiated from the original work by Brennan *et al.* (2003). The authors of this paper studied 604 patients who presented in emergency room with chest pain. The blood samples were collected at the presentation of the emergency room and the protein level of MPO (myloperoxidase) and other known cardiovascular biomarkers were measured. The patient's outcome (any cardiovascular event) was monitored for 6 months. The study showed the MPO to be a new biomarker for the prediction of MACE (major adverse cardiac events) risk in 30 days after the presentation of chest pain in emergency room with accuracy about 60%. Recently, the FDA (U.S. Food and Drug Development) approved the CardioMPO kit for measurement of MPO level (http://www.fda.gov/cdrh/reviews/K050029.pdf). In this paper, we introduce an application of the theory of linear predictive coding and its decision logic for feature extraction and classification of mass spectrometry signals in order to early predict patient's risk of major adverse cardiac events (Zhou et al. 2006). Applications of such computational frameworks have never been explored before for the analysis of proteomic data. We will show that the LPC model can provide a robust modeling of MS signals, which can be represented by LPC coefficients. We then show how the LPC vectors make it very convenient for classifying MS samples.

2 Feature Extraction of MS Data

It has been pointed out that digital signal processing can provide a set of novel and useful tools for solving highly relevant problems in genomics and proteomics (Anatassiou 2001, Vaidyanathan 2004). Recently, the applications of signal-processing based pattern analysis have been reported to be promising tools for the study of complex biological problems (Lazovic 1996, Wu & Castleman 2000, de Trad et al. 2002, Pham 2006). In this paper, we apply the principle of linear predictive coding (LPC) to extract the feature of mass spectrometry data, whose raw forms do not convey much information for the task of classification. The new MS feature can be represented by the LPC coefficients. The computation is based on the principle that the estimated value of a particular MS intensity value s_m at position or time n, denoted as $\hat{s}(n)$, can be calculated as a linear combination of the past p samples. This linear prediction can be expressed as (Makhoul 1975, Rabiner & Juang 1993)

$$\hat{s}(n) = \sum_{k=1}^{p} a_k \, s(n-k) \qquad (1)$$

where the terms $\{a_k\}$ are called the linear prediction coefficients (LPC), and p the number of poles.

The prediction error $e(n)$ between the observed sample $s(n)$ and the predicted value $\hat{s}(n)$ can be defined as

$$e(n) = s(n) - \hat{s}(n) = s(n) - \sum_{k=1}^{p} a_k \, s(n-k) \qquad (2)$$

From the above equation, it can be seen that the problem of linear prediction analysis we address herein is to optimally determine the set of predictor coefficients $\{a_k\}$ directly from the MS signal. Since the spectral properties of MS data can vary over time, the predictor coefficients at a given time n must be estimated from a short segment of the MS signal occuring around time n. Therefore, the solution is to find a set of predictor coefficients that minimize the mean-squared prediction error over a short segment of the whole MS signal.

A short-term MS signal, $s_n(m)$, and its error segment, $e_n(m)$, at time n can be defined as

$$s_n(m) = s(n+m) \qquad (3)$$

and

$$e_n(m) = e(n+m) \qquad (4)$$

The mean-squared error signal at time n to be minimized is defined as

$$E_n = \sum_m e_n^2(m) \qquad (5)$$

which can be expressed in terms of $s_n(m)$ as follows.

$$E_n = \sum_m \left[s_n(m) - \sum_{k=1}^{p} a_k s_n(m-k) \right]^2 \qquad (6)$$

Differentiating E_n, which is expressed in (6), with respect to each a_k and set the result to zero:

$$\frac{\partial E_n}{\partial a_k} = 0, \; k = 1, \ldots, p \qquad (7)$$

giving

$$\sum_m s_n(m-i) s_n(m) = \sum_{k=1}^{p} a_k \sum_m s_n(m-i) s_n(m-k) \qquad (8)$$

It can be noticed that the terms of the form $\sum s_n(m-i)s_n(m-k)$ are those of the short-term covariance of $s_n(m)$, that is

$$\phi_n(i,k) = \sum_m s_n(m-i) s_n(m-k) \qquad (9)$$

One possible way of defining the limits on m expressed in (9) is to assume that the segment, $s_n(m)$, is zero outside the interval $0 \leq m \leq N-1$, where N is the size of the short segment. This assumption is equivalent to that the signal $s(m+n)$ is multiplied by a finite length window, $w(m)$, which zero outside the range $0 \leq m \leq N-1$. Thus the segment for minimization can be expressed as

$$s_n(m) = \begin{cases} s(m+n)\, w(m) & : \; 0 \leq m \leq N-1 \\ 0 & : \; \text{otherwise} \end{cases} \qquad (10)$$

where $w(m)$ is usually a Hamming window.

Based on using the signal expressed in (10), the error signal $e_n(m)$ is exactly zero since $s_n(m) = 0$ for all $m < 0$, and for $m > N-1+p$ the prediction error is also zero because again $s_n(m) = 0$ for all $m > N-1$. Thus an optimal range of m used in defining the short segment of the sequence and the region over which the mean-squared error is minimized is from $m = 0$ to $m = N-1+p$ to minimize the errors at section boundaries. Using this range for m, the mean-squared error becomes (Rabiner & Juang 1993)

$$E_n = \sum_{m=0}^{N-1+p} e_n^2(m) \tag{11}$$

and $\phi_n(i,k)$ can be rewritten as

$$\phi_n(i,k) = \sum_{m=0}^{N-1+p} s_n(m-i)s_n(m-k), \tag{12}$$
$$1 \leq i \leq p, 0 \leq k \leq p$$

or

$$\phi_n(i,k) = \sum_{m=0}^{N-1-(i-k)} s_n(m)s_n(m+i-k), \tag{13}$$
$$1 \leq i \leq p, 0 \leq k \leq p$$

Since (14) is a function of $(i-k)$, the covariance function $\phi_n(i,k)$ can be reduced to the simple autocorrelation function:

$$\phi_n(i,k) = r_n(i-k) = \sum_{m=0}^{N-1-(i-k)} s_n(m)s_n(m+i-k) \tag{14}$$

Since the autocorrelation function is symmetric, that is $r_n(-k) = r_n(k)$, the system of LPC equations can be expressed as

$$\sum_{k=1}^{p} r_n(|i-k|)a_k = r_n(i),\ 1 \leq i \leq p \tag{15}$$

which describes a set of p equations in p unknowns, and can be expressed in matrix form as

$$\mathbf{R}\,\mathbf{a} = \mathbf{r} \tag{16}$$

where $\mathbf{R}$ is a $p \times p$ autocorrelation matrix (Toeplitz matrix which is symmetric with all diagonal elements being equal), $\mathbf{r}$ is a $p \times 1$ autocorrelation vector, and $\mathbf{a}$ is a $p \times 1$ vector of prediction coefficients:

$$\mathbf{R} = \begin{bmatrix} r_n(0) & r_n(1) & r_n(2) & \cdots & r_n(p-1) \\ r_n(1) & r_n(0) & r_n(1) & \cdots & r_n(p-2) \\ r_n(2) & r_n(1) & r_n(0) & \cdots & r_n(p-3) \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ r_n(p-1) & r_n(p-2) & r_n(p-3) & \cdots & r_n(0) \end{bmatrix}$$

$$\mathbf{a}^T = [\ a_1 \quad a_2 \quad a_3 \quad \cdots \quad a_p\ ]$$

and

$$\mathbf{r}^T = [\ r_n(1) \quad r_n(2) \quad r_n(3) \quad \cdots \quad r_n(p)\ ]$$

Thus, the LPC coefficients can be obtained by solving

$$\mathbf{a} = \mathbf{R}^{-1}\mathbf{r} \tag{17}$$

3 LPC-based Decision Logic

Let $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ be the vectors defined on a vector space V. A metric or distance d on V is defined as a real-valued function on the Cartesian product $V \times V$ if it has the properties of positive definiteness, symmetry, and triangle inequality. If a measure of dissimilarity satisfies only the property of positive definiteness, it is referred to as a distortion measure which is considered very common for the vectorized representations of signal spectra (Rabiner & Juang 1993).

In general, to calculate a distortion measure between two vectors $\mathbf{x}$ and $\mathbf{y}$, denoted as $D(\mathbf{x},\mathbf{y})$, is to calculate a cost of reproducing any input vector $\mathbf{x}$ as a reproduction of vector $\mathbf{y}$. Given such a distortion measure, the mismatch between two signals can be quantified by an average distortion between the input and the final reproduction. Intuitively, a match of the two patterns is good if the average distortion is small.

Consider the two spectra, magnitude-squared Fourier transforms, $S(\omega)$ and $S'(\omega)$ of the two signals s and s', where ω is the normalized frequency ranging from $-\pi$ to π. The log spectral difference between the two spectra is defined by (Rabiner & Juang 1993)

$$V(\omega) = \log S(\omega) - \log S'(\omega) \tag{18}$$

which is the basis for the distortion measure proposed by Itakura and Saito (IS) in their formulation of linear prediction as an approximate maximum likelihood estimation.

The Itakura-Saito distortion measure, D_{IS}, is defined as (Itakura & Saito 1970)

$$\begin{aligned} D_{IS} &= \int_{-\pi}^{\pi} [e^{V(\omega)} - V(\omega) - 1]\frac{d\omega}{2\pi} \\ &= \int_{-\pi}^{\pi} \frac{S(\omega)}{S'(\omega)}\frac{d\omega}{2\pi} - \log\frac{\sigma_\infty^2}{\sigma_\infty'^2} - 1 \end{aligned} \tag{19}$$

where σ_∞^2 and $\sigma_\infty'^2$ are the one-step prediction errors of $S(\omega)$ and $S'(\omega)$, respectively, and defined as

$$\sigma_\infty^2 \approx \exp\left\{\int_{-\pi}^{\pi} \log S(\omega)\frac{d\omega}{2\pi}\right\}. \tag{20}$$

It was pointed out that the Itakura-Saito distortion measure is connected with many statistical and information theories. A very useful distortion measure that is derived from the Itakura-Saito distortion measure is called the likelihood ratio (LR) distortion. The LR distortion measure, D_{LR}, is defined as (Rabiner & Juang 1993)

$$D_{LR} = \frac{\mathbf{a}'^T\,\mathbf{R}_s\,\mathbf{a}'}{\mathbf{a}^T\,\mathbf{R}_s\,\mathbf{a}} - 1 \tag{21}$$

where $\mathbf{R}_s$ is the autocorrelation matrix of sequence s associated with its LPC coefficient vector $\mathbf{a}$, and $\mathbf{a}'$ is the LPC coefficient vector of signal s'.

If the input (unknown) MS signal s_m is analyzed by the LPC which results in a set of LPC coefficients, then the spectral distortion between an unknown sample s_m and a particular known class i can be determined using the minimum rule as follows.

$$D_{min}(\mathbf{x}_m, \mathbf{c}^i) = \min_j D(\mathbf{x}_m, \mathbf{c}_j^i) \tag{22}$$

where D is a spectral distortion measure (if using the LR distortion then $D = D_{LR}$), $\mathbf{x}_m$ is the LPC vector of s_m, $\mathbf{c}_j^i$ is the LPC vector of the j sample that belongs to class i.

Table 1: Best and average sensitivity (SEN) and selectivity (SEL)

% Training	% SEN_{best}	% SEL_{best}	% SEN_{ave}	% SEL_{ave}
60	70.83	72.92	64.58	66.07
70	77.78	80.56	70.24	71.03
80	83.33	91.65	76.78	73.81
90	100	91.67	79.76	72.62

Using a simple decision logic, the unknown signal s_m is assigned to class i^* if the minimum distortion measure of its LPC vector $\mathbf{x}_m$ and the corresponding LPC vector $\mathbf{c}^i$ is minimum, that is

$$s_m \to i^*, \;\; i^* = \arg\min_i D_{min}(\mathbf{x}_m, \mathbf{c}^i) \qquad (23)$$

4 Experiment

We used high throughput, low resolution SELDI MS (www.ciphergen.com) to acquire the protein profiles from patients and controls. Figures 1 and 2 show the typical SELDI mass spectra of the control and MACE samples respectively. The protein profiles were acquired from 2 kDa to 200kDa. The design of the experiment originally described in (Zhou et al. 2006), and the result are presented as follows.

Control group (Zhou et al. 2006): This group has sixty patients who presented in emergency room with chest pain and the patients' troponin T test was consistently negative. These patients lived in the next 5 years without any major cardiac events or death. The total 166 plasma samples, 24 reference samples and 6 blanks were fractionated into 6 fractions using two 96-well plates containing anion exchange resin (Ciphergen, CA).

MACE group (Zhou et al. 2006): This group has 60 patients who presented in emergency room with chest pain but the patients' troponin T test was negative. However, the patients in this group had either a heart attack, died or needed revascularization in the subsequent 6 months. The blood samples used in this study were same as those used in (Brennan et al. 2003). Most new MPO data measured with FDA approved CardioMPO kit for these two groups are available – MPO levels for 56 (out of 60) patients in control group and 55 (out of 60) patients in MACE group are available. Statistical analysis shows that MPO alone can distinguish MACE from control with accuracy of better than 60%.

SELDI mass spectra: To increase the coverage of proteins in SELDI protein profiles, the blood samples were fractionated with HyperD Q (strong ion exchange) into 6 fractions. The protein profiles of fraction 1, 3, 4, 5 and 6 were acquired with two SELDI Chips: IMAC and CM10. There are a few different SELDI chips with different protein binding properties. General speaking, the more types of the SELDI chips are used, the more proteins are likely to be detected. However, due to the high concentration dynamic range of the proteins in human blood, the total number of proteins to be detected by the protocol we are using is very limited. We estimate that the number of the proteins we are able to detect is about one-thousand, while the total protein number in human blood is estimated to be tens of thousands. For example, MPO can be accurately measured with immunoassay (CardioMPO) but could not be detected with SELDI MS.

Number of MS spectra: MS spectra for each sample in each fraction was acquired in duplicate, so 120 samples (60 controls and 60 MACEs) in each fraction in one type of SELDI chip have 240 spectra. There are 5 fractions (Fraction 1, 3, 4, 5 and 6) and two types of SELDI chips (IMAC and CM10). Thus the total number of SELDI MS spectra to be analyzed is $240 \times 5 \times 2 = 2400$.

SELDI MS reproducibility (intensity measurement error): The reproducibility of the mass spectra was monitored with a pooled sample (12 samples were combined together to form a pooled sample) and total 24 spectra with the pooled sample were acquired at same time with all samples. The intensities of top 20 to 30 peaks in MS were compared and statistically analyzed. The estimated measurement error on peak intensity is about 20%-30%. The peak intensity is in the relative scale with the highest value of 100%. The relative peak intensity value is also dependent of the algorithms of baseline subtraction and normalization.

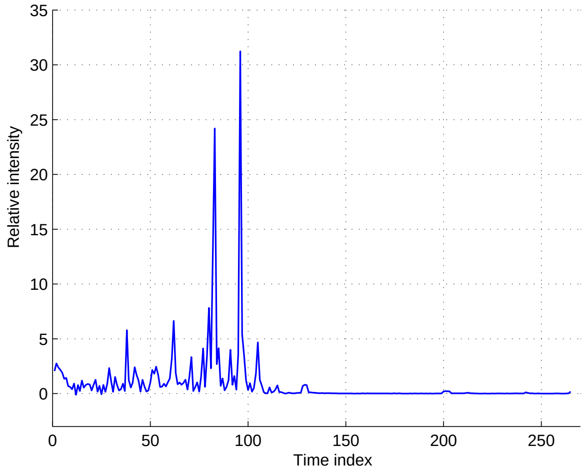


Figure 1: SELDI-MS control sample

Result: To focus our study on the prediction problem using SELDI mass spectra, we randomly selected two fractions to carry out the experiment. Because of the short length of the samples, we concatenated the corresponding samples of the two fractions for the extraction of the LPC coefficients. We estimated the number of poles p for the LPC analysis by using the semi-variograms of the mass spectra (see Figure 3) which reveal the number of poles being about 40 for the LPC analysis (Pham & Wagner 1998). The variogram is a function which expresses the spatial correlation of a regionalized variable (Deutsch 2002). In probabilistic notation, the variogram, $2\gamma(h)$, is defined as the expected value:

$$2\gamma(h) = E\{[s(i) - s(j)]^2\}, \; h_{ij} = h \qquad (24)$$

where h is a lag distance that separates $s(i)$ and $s(j)$.

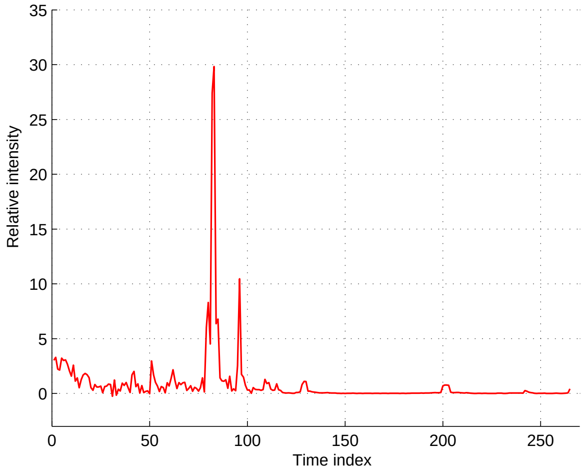


Figure 2: SELDI-MS MACE sample

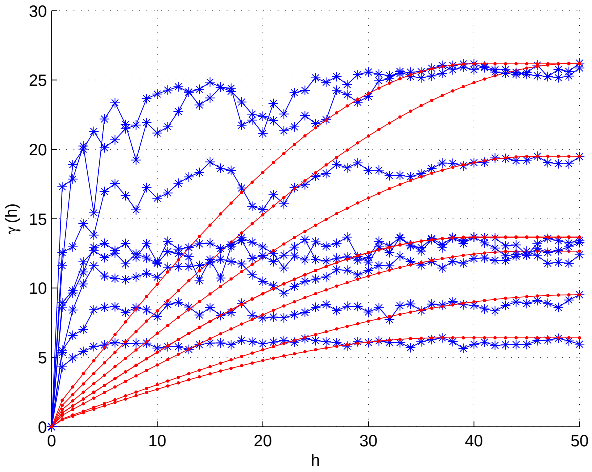


Figure 3: Variograms of SELDI-MS samples

The semi-variogram is half of the variogram, that is, $\gamma(h)$. The experimental semi-variogram for lag distance h is defined as the average squared difference of values separated by h:

$$\gamma(h) = \frac{1}{2N(h)} \sum_{(i,j)|h_{ij}=h} [s(i) - s(j)]^2 \qquad (25)$$

where $N(h)$ is the number of pairs for lag h.

The term $\gamma(h)$ in Figure 3 refers to the spatial variance of the MS relative intensities as the function of the lag distance h. The non-smooth curves are constructed using the experimental semi-variograms defined in (25); whereas the smooth curves are the theoretical semi-variograms generated by the spherical model which is defined as (Isaaks & Srivastava 1989)

$$\gamma(h) = \begin{cases} 1.5\frac{h}{a} - 0.5(\frac{h}{a})^3 & : \quad h \leq a \\ 1 & : \quad \text{otherwise} \end{cases} \qquad (26)$$

where a is called the range of the semi-variogram and can be considered to be an optimal number of poles p in the LPC analysis.

Using the leave-one-out method, we obtained the classification rate of 83.34%, where 99 out of 120 MACE samples and 101 out of 120 control samples were correctly classified. We then run different tests for p= 20, 25, 30, 35, 40, and 50 with different ratios of training and testing data to compute the sensitivity and selectivity of the classification. Sensitivity is the percentage of the MACE (diseased) samples that are correctly identified, whereas specificity is the percentage of the control (non-diseased) samples that are correctly identified. These results are shown in Table 1. In particular, the results are better when the numbers of poles are between 30 and 40, which are in agreement with the indication of the semi-variograms. In other applications such as speech recognition (Rabiner & Juang 1993), reasonable numbers of poles for the LPC analysis have been determined by experiences through training and testing of the speech recognizers. We present herein a useful way for selecting a good number of poles based on the theory of geostatistics provided that the samples are spatially correlated.

5 Conclusion

It has been predicted that the advancement of proteomics pattern diagnostics might represent a revolution in the field of molecular medicine, because this technology has the potential of developing a new model for early disease detection. The clinical impact of proteomic pattern diagnostics is still in the very early stage where the results have not been validated in large trials. Furthermore, recent research outcomes have illustrated the role of MS-based proteomics as an indispensable tool for molecular and cellular biology and for the emerging field of systems biology (Aebersold et al. 2003).

Given these promising results, identifying biomarkers using MS data is a challenging task, which requires the combination of the contrast fields of knowledge of modern biology and computational methodology. We have presented in this paper a novel application of a theory of linear predictive coding in signal processing for extracting robust features of mass spectrometry data that can be effectively utilized for the classification of MS spectra. The initial results using a small SELDI-MS dataset show the potential application of the proposed technique for predicting patient's major adverse cardiac risk and also indicate the potential use of the data for biomarker discovery.

References

Aebersold, R., & Mann, M. (2003), 'Mass spectrometry-based proteomics', *Nature* **422**, 198–207.

Anatassiou, D. (2001), 'Genomic signal processing', *IEEE Signal Processing Magazine* **18**, 8–20.

Anderle, M., Roy, S., Lin, H., Becker, C., & Joho, K. (2004), 'Quantifying reproducibility for differential proteomics: noise analysis for protein liquid chromatography-mass spectrometry of human serum' , *Bioinformatics* **20**, 3575–3582.

Ball, G., Mian, S., Holding, F., Allibone, R.O., Lowe, J., Ali, S., Li, G., McCardle, S., Ellis, I.O., Creaser, C., & Rees, R.C. (2002), 'An integrated approach utilizing artificial neural networks and SELDI mass spectrometry for the classification of human tumours and rapid identification of potential biomarkers', *Bioinformatics* **18**, 395–404.

Brennan, M.-L., Penn, M.S., Van Lente, Nambi, V., Shishehbor, M.H., Aviles, R.J., Goormastic, M., Pepoy, M.L., McErlean, E.S., Topol, E.J., Nissen, S.E., & Hazen, S.L. (2003), 'Prognostic value of myeloperoxidase in patients with chest pain', *The New England Journal of Medicine* **13**, 1595–1604.

Conrads, T.P., Zhou, M., Petricoin III, E.F., Liotta, L. & Veenstra, T.D. (2003), 'Cancer diagnosis using proteomic patterns', *Expert Rev. Mol. Diagn.* **3**, 411–420.

Deutsch, C.V. (2002), *Geostatistical Reservoir Modeling.* Oxford University Press, New York.

de Trad, C.H., Fang, Q. & Cosic, I. (2002), 'Protein sequence comparison based on the wavelet transform approach', *Protein Engineering* **15**, 193–203.

Gray, R.M. (1984), 'Vector quantization', *IEEE ASSP Mag.* **1**, 4–29.

Griffin, T., Goodlett, T. & Aebersold, R. (2001), 'Advances in proteomic analysis by mass spectrometry', *Curr. Opin. Biotechnol.* **12**, 607–612.

Isaaks, E.H. & Srivastava, R.M. (1989), *An Introduction to Applied Geostatistics.* Oxford University Press, New York, 1989.

Itakura, F. & Saito, S. (1970), A statistical method for estimation of speech spectral density and formant frequencies', *Electronics and Communications in Japan* **53A**, 36–43.

Lazovic, J. (1996), 'Selection of amino acid parameters for Fourier transform-based analysis of proteins', *CABIOS* **12**, 553–562.

Levner, I. (2005), 'Feature selection and nearest centroid classification for protein mass spectrometry', *BMC Bioinformatics* **6:68**, (http://www.biomedcentral.com/1471-2105/6/68).

Lilien, R.H., Farid, H., & Donald, B.R. (2003), 'Probabilistic disease classification of expression-dependent proteomic data from mass spectrometry of human serum', *J. Computational Biology* **10**, 925–946.

Linde, Y., Buzo, A., and Gray, R.M. (1980), 'An Algorithm for Vector Quantization', *IEEE Trans. Communications* **28**, 84–95.

Makhoul, J. (1975), 'Linear prediction: a tutorial review', *Proc. IEEE* **63**, 561–580.

Morris, J.S., Coombes, K.R., Koomen, J., Baggerly, K.A., & Kobayashi, R. (2005), 'Feature extraction and quantification for mass spectrometry in biomedical applications using the mean spectrum', *Bioinformatics* **21**, 1764–1775.

Petricoin, E.F., et al. (2002), 'Use of proteomic patterns in serum to identify ovarian cancer', *Lancet* **359**, 572–577.

Petricoin, E.F. & Liotta,L.A. (2003), 'Mass spectrometry-based diagnostics: The upcoming revolution in disease detection', *Clinical Chemistry* **49**, 533–534.

Pham, T.D. & Wagner, M. (1998), 'A geostatistical model for linear prediction analysis of speech', *Pattern Recognition* **31**, 1981–1991.

Pham, T.D. (2006), 'LPC cepstral distortion measure for protein sequence comparison', *IEEE Trans. NanoBioscience* **5**, 83–88.

Rabiner, L.R., Sondhi M.M., and Levinson, S.E. (1984), 'A vector quantizer incorporating both LPC shape and energy', *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, pp. 17.1.1–17.1.4,.

Rabiner, L. & Juang, B.H. (1993), *Fundamentals of Speech Recognition.* New Jersey, Prentice Hall.

Salmi, J., Moulder, R., Filen, J.-J., Nevalainen, O.S., Nyman, T.A., Lahesmaa, R. & Aittokallio, T. (2006), 'Quality classification of tandem mass spectrometry data', *Bioinformatics* **22**, 400–406.

Sauter, E., et al. (2002), 'Proteomic analysis of nipple aspirate fluid to detect biologic markers of breast cancer', *Br. J. Cancer* **86**, 1440–1443.

Shin, H. & Markey, M.K. (2006), 'A machine learning perspective on the development of clinical decision support systems utilizing mass spectra of blood samples', *J. Biomedical Informatics* **39**, 227–248.

Sorace, J.M. & Zhan, M. (2003), 'A data review and re-assessment of ovarian cencer serum proteomic profiling', *BMC Bioinformatics* **4:24**, (http://www.biomedcentral.com/1471-2105/4/24).

Tibshirani, R., Hastie, T., Narasimhan, B., Soltys, S., Shi, G., Koong, A. & Le, Q.-T. (2004), 'Sample classification from protein mass spectrometry, by 'peak probability contrasts'', *Bioinformatics* **20**, 3034–3044.

Vaidyanathan, P.P. (2004), 'Genomics and proteomics: A signal processor's tour', *IEEE Circuits and Systems Magazine*, Fourth Quarter pp. 6–28.

Xiong, J. (2006), *Essential Bioinformatics*, Cambridge University Press, New York.

Yu, J.S., Ongarello, S., Fiedler, R., Chen, X.W., Toffolo, G., Cobelli, C. & Trajanoski, Z. (2005), 'Ovarian cancer identification based on dimensionality reduction for high-throughput mass spectrometry data', *Bioinformatics* **21**, 2200–2209.

Weir, M.P., Blackstock, W.P., & Twyman, M. (2003), Proteomics, *in* C.A. Orengo, D.T. Jones, and J.M. Thornton, eds, 'Bioinformatics: Genes, Proteins & Computers', BIOS Scientific Publishers, pp. 245–257.

Wu, Q., & Castleman, K.R. (2000), 'Automated chromosome classification using wavelet-based band pattern descriptors', *Proc. 13th IEEE Symp. Computer-Based Medical Systems*, pp. 189–194.

Wu, B., Abbott, T., Fishman, D., McMurray, W., Mor, G., Stone, K., Ward, D., Williams, K. & Zhao, H. (2003), 'Comparison of statistical methods for classification of ovarian cancer using mass spectrometry data', *Bioinformatics* **19**, 1636–1643.

Wulfkuhle, J.D., Liotta, L.A. & Petricoin, E.F. (2003), 'Proteomic applications for the early detection of cancer', *Nature* **3**, 267–275.

Zhou, X., Wang, H., Wang, J., Hoehn, G., Azok. J., Brennan, M.L., Hazen, S.L., Li, K., & Wong, S.T.C., (2006), 'Biomarker discovery for risk stratification of cardiovascular events using an improved genetic algorithm', *Proc. IEEE/NLM Int. Symposium on Life Science and Multimodality*, July 13-14, Washington, DC.

Automated Sub-Cellular Phenotype Classification: An Introduction and Recent Results

N. Hamilton[1,2,3] **R. Pantelic**[1,2] **K. Hanson**[1,2] **J.L. Fink**[1,2]
S. Karunaratne[1,2] **R.D. Teasdale**[1,2]

[1]Institute for Molecular Bioscience
[2]ARC Centre in Bioinformatics
[3]Advanced Computational Modelling Centre
The University of Queensland, Brisbane Qld 4072, Australia
Email: n.hamilton@imb.uq.edu.au

Abstract

The genomic sequencing revolution has led to rapid growth in sequencing of genes and proteins, and attention is now turning to the function of the encoded proteins. In this respect, microscope imaging of a protein's subcellular location is proving invaluable. High-throughput methods mean that it is now possible to capture images of hundreds of protein localisations quickly and relatively inexpensively, and hence genome-wide protein localisation studies are becoming feasible. However, to a large degree the analysis and localisation classification are still performed by the slow, coarse-grained and possibly biased process of manual inspection. As a step towards dealing with the fast growth in subcellular image data the Automated Sub-cellular Classification system (ASPiC) has been developed: a pipeline for taking cell images, generating statistics and classifying using SVMs. Here, the pipeline is described and correct classification rates of 93.5% and 86.5% on two 8-class subcellular localisation datasets are reported. In addition we present a survey of other important applications of cell image statistics. The complete image sets are being made available with the aim of encouraging further research into automated cell image analysis and classification.

Keywords: Subcellular phenotype, subcellular localisation, image statistics, image classification, machine learning.

1 Introduction

The advent of fast, automated and inexpensive sequencing technologies led to the completion of the human, mouse and many other genomes, and an exponential growth in genomic data. Sequence-based machine learning has played a pivotal role in automated annotation and prediction of structure and function of novel sequences and has become an essential tool. However, while sequence data are invaluable further information, such as experimentally-determined subcellular localisation (see Figure 1), trafficking and interaction partners is required to fully understand the functions of the tens of thousands of proteins that have been identified (Fink, Aturaliya, Davis, Zhang, Hanson, Teasdale & Teasdale 2006)(Stow & Teasdale 2005). Sequence-based approaches have been applied to predicting localisation (Yu, Chen, Lu & Hwang 2006) but tend to need high homology to proteins of known localisation, and so experimental verification is a necessity. Automated fluorescent microscope imaging technologies mean that it is now possible to capture hundreds of images per second including multiple fluorophores for cells under a variety of experimental conditions (Lang, Yeow, Nichols & Scheer 2006)(Bonetta 2005). Furthermore, cells may now be imaged in 3D, or indeed in 4D with a 3D stack captured over time to observe protein trafficking in real time. The desire and the ability to do high-throughput screenings of protein localisation and trafficking is leading to a rapid growth in cell images in need of analysis on a scale comparable to that of the genomic revolution. Automated image analysis and classification is essential.

Much of the reason for the rapid growth of machine learning techniques applied to genomic data is ubiquitousness of sequence information from publicly available databases. Until recently, dissemination of cell image data involved selection of a few "representative" images for publication in a paper. But a much richer range of data is becoming available with large-scale publicly accessible cell image databases such as the LOCATE mouse protein subcellular localisation database (Fink et al. 2006) (more databases are listed in (Matthiessen 2003)). Currently, the databases are largely human-curated, but the data becoming available offer many opportunities to train and apply machine learning techniques to experimental image classification and analysis. There is a real need to refine, discriminate and quantify to produce annotation of images in cell databases. Cells can exhibit a wide range of behaviours over the cell cycle that can potentially skew results, and techniques have been developed to automatically determine the phase of cell image sequences (Pham, Tran, Zhou & Wong 2006). Aberrant cell morphology also presents an interesting challenge to image classification. Atypical morphology may skew data when examining normal cells. Alternatively, atypical morphology may be the primary attribute which assists in the discrimination between, for instance, normal cells and cancerous cells (Thiran, Macq & Mairesse 1994). On the quantitative side, methods have been developed to select and count substructures, such as puncta (Pham, Crane, Tran & Nguyen 2004), from cell images. These automated techniques offer the opportunity to annotate at a much more refined level, thereby increasing the quality of data and allowing more subtle hypotheses to be tested.

As well as presenting the ASPiC pipeline, the aim here is to draw attention to the large image data sets that are now becoming available and to the great need for, and applications of, machine learning to these sets. In the following, we begin with an introduction to image statistics and their potential applications to

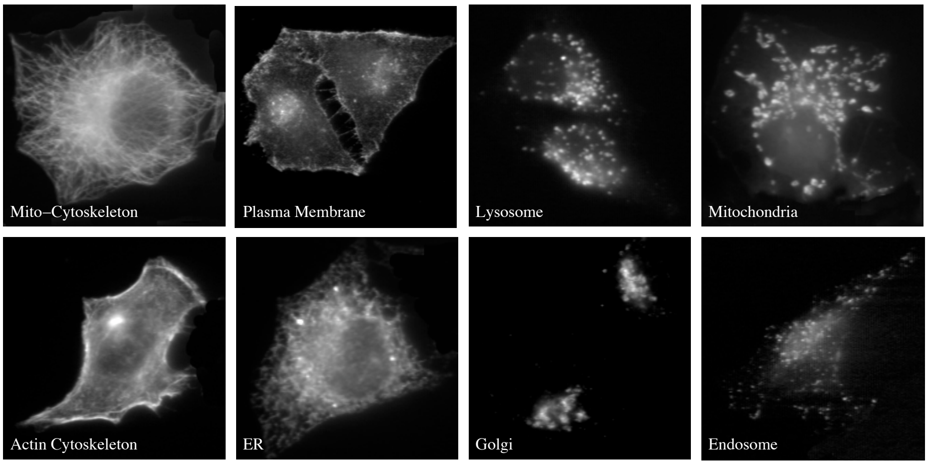


Figure 1: Samples of endogenously expressed proteins from our datasets

high-throughput cell imaging problems. The ASPiC system and the image sets being made available are then described, and we conclude with some remarks on future directions for analysis of subcellular imaging.

2 Image Statistics and their Uses

A common problem in cell biology is to determine the subcellular localisation of a given protein: does it localise to the nucleus or the cytoplasm? Has treatment of the cell modulated the localisation of an individual protein? Examples of fluorescently-tagged proteins exhibiting various subcellular localisations are shown in Figure 1. While applying learning algorithms with the image itself as input has proved quite successful (Danckaert, Gonzalez-Couto, Bollondi, Thompson & Hayes. 2002), generation of numeric image measures has a wider range of applications. The aim is to find measures that can differentiate between localisations in distinct classes when localisations within a given class can exhibit a very wide range of expression patterns and morphologies. To be applicable to as wide a range of images as possible, cell image measures should ideally be invariant under rotation, translation and scale changes. Here, some of the measures that have been applied to quantifying subcellular localisation images are described. More may be found in (Conrad, Erfle, Warnat, Daigle, Lorch, Ellenberg, Pepperkok & Eils 2004) and (Huang & Murphy 2004).

Area Intensity Measures

Typically, for subcellular localisation a pair of microscope images will be taken: one of the fluorescently-tagged protein of interest (POI); and one in which the nucleus of the cell is fluorescently-labelled. From these, image masks of the POI and the nucleus are created (see Figure 2). Area and intensity measures may then be calculated: the area of the region that the POI is expressed in; average intensity across the masked region; the ratio of the intensities of the POI in the nuclear to non-nuclear regions; area and intensity averages over various intersections and differences of POI and nuclear masks; and the standard deviation of the POI image intensity in the mask region. Statistics such as these will easily differentiate between proteins expressing in the cytoplasm and the nucleus. Generally, area and intensity *ratio* measures are better in that they are less affected by the image resolution or exposure.

Haralick Texture Measures

A more refined set of image measures that have been applied to a wide range of problems such as satellite imaging and computerized tomography are the Haralick texture measures (Haralick 1979). The idea is to find the correlation (and other measures) between pixel intensities at a given distance and angle. Hence, if an image contained a series of high-intensity bands at a given separation, a Haralick correlation measure (with the appropriate distance and angular separation) will return a high value. Suppose an image contains N gray tones, then for a given pixel pair separation d and angle θ a $N \times N$ *gray tone co-occurrence table* P is constructed. The entries P_{ij} are the relative frequency with which two pixels separated by distance d and angle θ have gray tone values i and j, respectively. Measures such as *uniformity*: $\sum_{ij} P_{ij}^2$; *entropy*: $\sum_{ij} P_{ij} \log P_{ij}$; and *correlation*: $\sum_{ij} (i-\mu)(j-\mu)P_{ij}/\sigma^2$, where μ and σ are the mean and standard deviation of the pixel intensities, are then applied to the occurrence matrix. The Haralick statistics may be applied to the whole of the mask region of the POI, or to subregions of it defined by intersections and differences of the POI and nuclear masks. Since there are many possible choices of d and θ, for a given d the occurrence matrix is sometimes averaged over a range of values of θ such as 0°, 90°, 180° and 270° degrees. This has the advantage of reducing rotational variance, though may lead to a reduced signal. More Haralick measures are given in the Appendix.

Zernike Moments

Another set of measures that are computationally relatively inexpensive and have proved useful in cell imaging are the Zernike moments (Khotanzad & Hong 1990)(Boland, Markey & Murphy 1998)(Zernike 1934). These are calculated using an orthogonal polynomial set, the Zernike Polynomials, on the unit circle. Given a complete (infinite) set of Zernike moments for a given image it is in theory possible to reconstruct the image perfectly. However, calculation of the first few moments will often give a general sense of the morphology of the imaged object, much as a small subset of Fourier coefficients will for a time series (Boland et al. 1998). The discrete equations for the Zernike moments are given in the Appendix.

Applications

In general, no one statistic is a good predictor of subcellular phenotype, and so machine learning techniques such as neural networks and support vector machines have been applied to classification based on image statistics. As shown in the next section, classification accuracies of greater than 90% may be obtained. In addition to allowing high-throughput classification of new images, an immediate application of a phenotype classifier is to *image database curation.* As the size and number of image databases expands, quality and uniformity of human classification becomes an issue. Experiments by Murphy lab on a set of images with 10 distinct known subcellular localisations (similar to those in Figure 1) found human classifiers had an accuracy of 83% compared to 92% for a machine classifier (Huang & Murphy 2004)(Murphy, Velliste & Porreca 2003). This may in part be explained by the inherent difficulty in providing accurate classifications for hundreds of images over a long time period, but it is worth noting that the human eye only registers a few tens of distinct gray scale values at a time, while a 8-bit cell image file may have close to 250, and hence there is potential for software to "see" much more. By flagging for re-examination the images for which the human and machine classifications disagree, there is the potential to significantly improve database quality.

Other applications of image statistics include *representative image selection* and *statistical comparison* of image sets. In the former, given a set of images of a particular protein, the aim is to select the image that best represents the variety of distributions observed. This may be done by finding the image that has statistics closest to the mean statistics vector of all the images (Roques & Murphy 2002). In the latter, there are two sets of experimental conditions for a protein where it is required to ascertain whether the two distributions are statistically significantly different. Using the Hotelling T^2-test on the image statistics, it has been shown (Roques & Murphy 2002)(Huang & Murphy 2004) that sets with the same localisation may be correctly identified, and that expression patterns that were known to be different can be distinguished, even to the extent of differentiating visually-indistinguishable images.

Finally, cell image statistics offer the possibility of searching image databases for similar images on the basis of image content rather than the (possibly biased) keywords supplied by the experimenter. Arguably the most powerful tool for genomic inference is the BLAST sequence matching algorithm (Altschul, Gish, Miller, Myers & Lipman 1990) that finds and quantifies similarity between sequences in a database. Once an "image BLAST" is developed for cell image databases, the ability to deduce biological inferences and associations will be greatly increased.

3 The ASPiC Pipeline

The Automated Subcellular Phenotype Classification system (ASPiC) is a fully-automated pipeline from experimental image to a subcellular classification suitable for direct database entry. The principle steps are outlined in Figure 2 and are described in detail below. ASPiC is currently being integrated into the LOCATE database (Fink et al. 2006) where it is providing classification on a 3-class nuclear or cytoplasm or nuclear and cytoplasm problem. Here, we describe its application to two 8-class subcellular localisation datasets. Other applications such as representative image selections are under development. The major parts of ASPiC are implemented in C++ using the ImageMagick++ image libraries.

3.0.1 Image Sets

An image collection was created for each of 8 subcellular organelles in two types of sets; one in which an *endogenous* protein or feature of the specific organelle was detected with a fluorescent antibody or other probe; and another in which an epitope- or fluorescently-tagged protein was transiently expressed (transfected) in the specific organelle and subsequently detected. Each set consisted of 50 images. Each image was accompanied by an additional image of the cells counterstained with the DNA specific dye 4,6-diamidino-2-phenylindole (DAPI), which highlights the location of the nucleus of every cell in the image. All images were of fixed HeLa cells, taken at 60X magnification under oil immersion. More details are available with the image sets.

Automated Cropping and Cell Selection

The first step is to select regions in which proteins are expressing in the POI and nuclear images using an automated grayscale thresholding scheme. A variety of schemes were tried, but the best was found to be to choose a minimum intensity and a maximum intensity, and take the average (μ) and standard deviation (σ) of the pixels with intensities in this range. For the POI images, a minimum intensity of 30 and a maximum of 250 is used, and 20 and 250 for the nuclear images. The threshold for the image is then set to $\mu - 0.9\sigma$, and above threshold pixels define the regions of interest. Using these minima and maxima in the calculation of μ and σ removes pixels that are either certainly background or overexposed, and gives results that were generally in agreement with the regions that the eye considers to be of interest. Contiguous regions are then selected and cropped in the POI image, together with the corresponding area in the nuclear image. To remove artefacts, any selected region that is small or faint is discarded. The circularity (perimeter squared over area) of the nuclear region mask is calculated, and nuclei with large circularity are discarded as these usually represent multiple or poorly imaged nuclei in a cell. In the case of multiple nuclei in the cropped region, the most central is selected. In some cases cells are not separable by thresholding. This is detected by multiple disjoint nuclei (in the nuclear mask) being contained within the POI mask, and is treated as a single cell by ASPiC. Using these criteria approximately 93% of source images are found to contain one or more valid cells.

Image Statistics

For each cropped cell a total of 95 statistics are generated composed of 25 area and intensity measures, 21 Haralick statistics and 49 Zernike moments of up to degree 12. Since Zernike moments are not rotationally invariant, the magnitudes of the moments are taken to minimise sensitivity to cell orientation. The area and intensity measures arise from taking intersections and differences of the POI and nuclear masks, and calculating average intensities, areas, intensity ratios and area ratios. Haralick measures were chosen from a list of those shown to be good for distinguishing subcellar localisation in (Conrad et al. 2004). Details of the measures used by ASPiC may be found in the Appendix.

Training and Testing

Support vector machine classifiers were created for the 8-class endogenous and 8-class transfected image

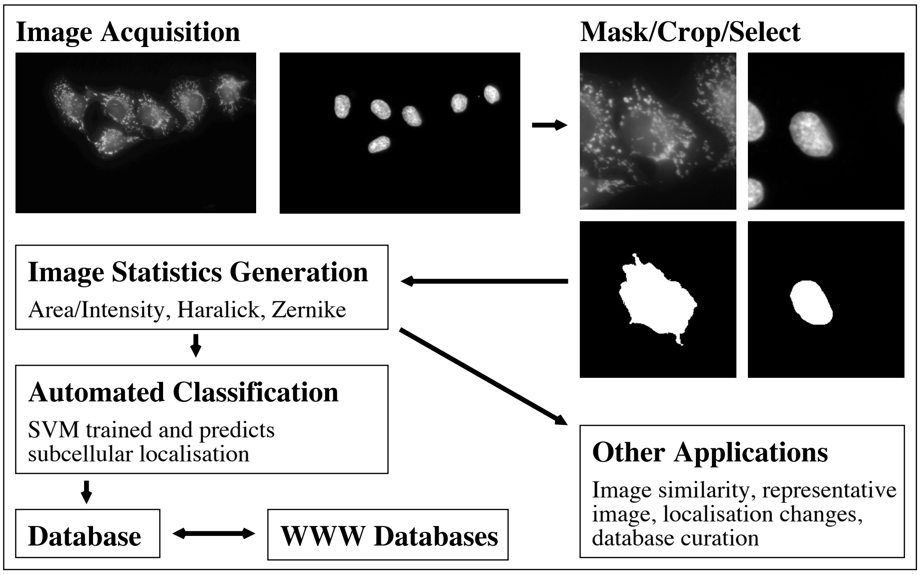


Image Acquisition
Mask/Crop/Select
Image Statistics Generation
Area/Intensity, Haralick, Zernike
Automated Classification
SVM trained and predicts subcellular localisation
Database
WWW Databases
Other Applications
Image similarity, representative image, localisation changes, database curation

Figure 2: The Automated Subcellular Phenotype Classification System (ASPiC)

sets using the libsvm software with an RBF kernel (Chang & Lin 2001). An ANN was also briefly trained and tested on the same data sets, but found to give lower performance (data not shown). Two parameters are required to create and train the SVM: γ, the coefficient of the exponent for the RBF kernel and C, the penalty parameter of the error term. A grid search was performed to choose the values of γ and C that gave the best 5-fold cross validated performance on each data set. On the endogenous data set, the best cross validation accuracy was 94.3% using $\gamma = 0.03716$ and $C = 26.91$. For the tranfected data set, an accuracy of 89.8% was obtained using $\gamma = 0.03284$ and $C = 89.84$. Linear kernels were also tested with 5-fold cross validation on each data set, and gave 91.8% and 89.2% on the endogenous and transfected data sets, respectively. Polynomial kernels were also tested, but were also found not to perform as well as the RBF kernel. Once the RBF kernel and parameters were fixed as above, 100 random (class-balanced) splits of the data into 4/5 training and 1/5 testing set were performed and an SVM trained and tested. For each test set, the overall percentage of correct predictions was recorded, as well as the percentage of correct predictions for each class of the data in the test set.

Schemes for selecting subsets of the statistics ranked by F-score were also investigated using the *fselect* script available for libsvm. Selection by F-scores has been shown to significantly improve performance on some data sets (Chen & Lin 2006). The F-scores varied widely, from 0.02 to 9.2 for the endogenous data set, and from 0.04 to 6.2 on the transfected data set. There was no clear bias in ranking either subregion, Zernike or Haralick statistics highly, with all three types represented in the top ten. For each of the endogenous and transfected data sets, 5-fold cross validating with the best ranked subsets of 95, 72, 47, 23, 11, 5 and 2 features showed either no improvement or significantly degraded performance.

Post Classification Filtering

ASPiC includes a voting system for multiple classifications of the same protein in distinct cells where split votes are broken by the maximum confidence score output by the SVM. There are a variety of approaches to post-classification filtering (Chen, & Murphy 2006), however, here we report raw classification accuracies in order that the true accuracy may be seen.

3.0.2 Classification Accuracy and Comparing the Incomparable

Over 100 trials of splitting data sets 4/5 to 1/5 for training/testing the average correct classification rates were 93.1% for the endogenous test sets and 87.9% for the transfected, with standard deviations of 1.58 and 2.51, respectively. To test which classes were accurately or poorly classified, the classification accuracies for each image class were also recorded and the averages are given in Table 1. Generally, the classification accuracies are high, though certain classes such as the cytoskeleton classes are less well-predicted for transfected cells. Those that are poorly predicted tend to be those that are visually similar to other classes. ASPiC has also been tested using only those statistics that require a POI and not a nuclear image, and gave cross-validation results around 2.5% lower than those above. Before comparing these results with previous literature, it should be made clear that each group is testing their system on distinct image sets with different numbers of subcellular classes and varying degrees of automation, and hence are not necessarily directly comparable. Murphy lab have been developing and improving subcellular phenotype classifiers for a number of years and have contributed much in the area of subcellular image statistics and their uses. The most recent report (Huang & Murphy 2004) on subcellular classification gives 88% for a pure neural network classifier on a 10-class problem with manual cropping and curating. The image sets were prepared using protein antibodies of known localisation, and hence are comparable to our endogenous image sets. Using a majority voting system combining a number of learning algorithms, 92% was obtained on the same set. A wide range of statistics were used and various feature selection algorithms used to select the best for training including Zernike moments and Haralick measures, though their implementation of Haralick measures differs from ASPiC's in that ASPiC does not average over a range of angles. Also of interest is the work of Conrad et al. (Conrad et al. 2004) in which a wide variety of image statistics, feature selection and learning algorithms were

Endo.	Mito-Cyto.	Endosome	ER	Golgi	Actin-Cyto.	Lysosome	Mitochondria	PM
#	45	31	59	48	29	62	68	22
Acc.	96.7	93.5	93.9	98.9	83.6	94.9	91.3	88.8
Trans.	Cytoplasm	Lysosome	ER	Endosome	Peroxisome	Mito-Cyto.	Actin-Cyto.	Nucleus
#	43	16	59	30	34	37	27	23
Acc.	99.7	98.7	84.5	80.4	89.0	78.5	85.5	100

Table 1: Average classification percentages on Endogenous and Transfected test sets over 100 randomised splits of the data into 4/5 training, 1/5 testing.

tested on 11 classes of subcellular phenotype images. Of the methods tested, they found stepwise feature selection in conjunction with a SVM offered the best performance with an accuracy of 82.2%. While comparison is problematic, ASPiC is certainly competitive with a 93.1% accuracy, it is simple in that it is fully-automated and uses a single machine learning method, and has been shown to perform well on uncurated images.

4 Conclusions

It is clear that image statistics can differentiate subcellular localisation to a high degree of accuracy, and that automation offers many advantages in high-throughput, time saved, consistency and quantification.

Currently, statistics are relatively slow to compute. Cells need to be selected from images of plates, cropped, and then up to a hundred statistics calculated, all of which can take of the order of seconds on a standard PC. When faster statistics are developed the range of applications will grow. One application would be to flow cytometry where cells are imaged and sorted on the fly (Bonetta 2005). With current technology, cells are typically sorted according to whether a cell is expressing a protein (bright) or not (dark). A fast classifier would enable selection of, for instance, all those cells for which a given protein is expressing in the Golgi, and then perform further experiments on those. New statistics we are developing look promising as quick, relatively accurate measures with no cropping.

As the flood of cell image data begins, the need for new applications of classification and discrimination are greatly increasing. Certainly there is a need for automated classification, but cell image databases also need the ability to be *queried by image example* in a way that understands the content of the image rather than by matching researcher-supplied keywords. If a researcher was looking to see if a protein localised to the Golgi, they may not have noted that it was in fact localising to a subregion of the Golgi. However, an image content-based search might provide that level of discrimination. In the future, as biological databases become more integrated and queryable, it should be possible, for instance, with a few mouse clicks to start with a protein sequence, find images of its subcellular localisation, "image BLAST"to find proteins that exhibit similar expression or co-expression patterns, then read source literature on the proteins.

Experimental images described herein are available via the LOCATE web interface (Fink et al. 2006).

5 Appendix: Features used in ASPiC

Haralick Texture Measures

Suppose an image contains N gray tones, then for a given pixel pair separation d and angle θ a $N \times N$ *gray tone co-occurrence table* P is constructed. The entries P_{ij} are the relative frequency with which two pixels separated by distance d and angle θ have gray tone values i and j, respectively. This definition of the gray tone co-occurrence table is as in (Haralick 1979) with the minor variation that the matrix has been normalised to give relative frequencies rather than counts of pixel pairs. The following image statistics are then calculated in ASPiC.

Correlation: $\sum_{ij} (i-\mu)(j-\mu)P_{ij}/\sigma^2$ where μ and σ are the mean and standard deviation of the pixel intensities.
$d = 3$, $\theta = 0$; $d = 4$, $\theta = 45$; $d = 3$, $\theta = 135$.

Correlation2: Haralick's second information measure of correlation. See (Haralick, Shanmugam & Dinstein 1973).
$d = 2$, $\theta = 0$; $d = 3$, $\theta = 45$; $d = 1$, $\theta = 135$.

Contrast: $\sum_{ij}(i-j)^2 P_{ij}$
$d = 5$, $\theta = 0$; $d = 5$, $\theta = 135$.

Inverse difference moment: $\sum_{ij} P_{ij}/(1+(i-j)^2)$
$d = 1$, $\theta = 90$.

Unformity: $\sum_{ij} P_{ij}^2$
$d = 1$, $\theta = 0$; $d = 2$, $\theta = 0$; $d = 4$, $\theta = 45$.

Entropy: $\sum_{ij} P_{ij} \log P_{ij}$
$d = 4$, $\theta = 135$.

Sum entropy: $\sum_k ((\sum_{i+j=k} P_{ij}) \log(\sum_{i+j=k} P_{ij}))$
$d = 1$, $\theta = 0$; $d = 4$, $\theta = 90$.

Difference entropy: $\sum_k ((\sum_{|i-j|=k} P_{ij}) \log(\sum_{|i-j|=k} P_{ij}))$
$d = 4$, $\theta = 0$; $d = 3$, $\theta = 45$; $d = 1$, $\theta = 45$.

Sum variance: $\sum_k (k-S)^2 \sum_{i+j=k} P_{ij}$ where S is the sum entropy.
$d = 4$, $\theta = 90$.

Zernike Moments

The magnitudes of the first 12 Zernike moments are calculated exactly as described by the equations in (Boland et al. 1998) to give 49 features as follows. Let $I(x,y)$ be the pixel intensity at position (x,y). Define

$$Z_{nl} = \frac{n+1}{\pi} \sum_{x,y} V_{nl}^*(x,y) I(x,y)$$

where $x^2 + y^2 \leq 1$, $0 \leq l \leq n$, $n-l$ even, and V_{nl}^* is the complex conjugate of the Zernike polynomial of degree n and angular dependence l, given by

$$V_{nl}(x,y) = \sum_{m=0}^{(n-l)/2} \frac{(-1)^m (x^2+y^2)^{n/2-m} e^{il\theta} (n-m)!}{m!(\frac{n-2m+l}{2})!(\frac{n-2m-l}{2})!}$$

where $\theta = \tan^{-1}(y/x)$.

Cell images are centered when cropped. To scale each cell image into the unit circle, pixel coordinates are divided by 100 before calulation of the Zernike moments.

Subregion Statistics

Denoting the mask selected region of the POI by P, nuclear mask selected region by N, and the area of a region A by $|A|$, the following area statistics are calculated: $|P|$, $|N|$, $|P-N|$, $|N-P|$, $|P\cap N|$, $|N|/|P|$, $|P-N|/|N|$, $|P-N|/|P|$, $|N-P|/|P|$, $|N\cap P|/|P|$, $|N-P|/|N|$ and $|N\cap P|/|N|$. The variance over the POI mask region, as well as the ratio of the perimeter squared over the area of the POI mask region, are also calculated.

Denoting the average pixel intensity over a region A by $(A)_I$, the following intensity measures are calculated in the POI image: $(P)_I$, $(N)_I$, $(P-N)_I$, $(N-P)_I$, $(P\cap N)_I$, $(P-N)_I/(P)_I$, $(N-P)_I/(P)_I$, $(N\cap P)_I/(P)_I$, $(N-P)_I/(P)_I$, $(N\cap P)_I/(P)_I$, $(N)_I/(P)_I$ and $(P-N)_I/(N)_I$.

References

Altschul, S., Gish, W., Miller, W., Myers, E. & Lipman, D. (1990), 'Basic local alignment search tool', *J. Mol. Biol.* **215**, 403410.

Boland, M., Markey, M. & Murphy, R. (1998), 'Automated recognition of patterns characteristic of subcellular structures in fluorescence microscopy images', *Cytometry* **33**(3), 366–375.

Bonetta, L. (2005), 'Flow cytometry smaller and better', *Nature Methods* **2**, 785 – 795.

Chang, C.-C. & Lin, C.-J. (2001), *LIBSVM: a library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/ cjlin/libsvm`.

Chen, S.-C., & Murphy, R. F. (2006), 'A graphical model approach to automated classification of protein subcellular location patterns in multi-cell images', *BMC Bioinformatics* **7**(90).

Chen, Y.-W. & Lin, C.-J. (2006), *Feature extraction, foundations and applications*, Springer, chapter Combining SVMs with various feature selection strategies.

Conrad, C., Erfle, H., Warnat, P., Daigle, N., Lorch, T., Ellenberg, J., Pepperkok, R. & Eils, R. (2004), 'Automatic identification of subcellular phenotypes on human cell arrays', *Genome Research* **14**(6), 1130–6.

Danckaert, A., Gonzalez-Couto, E., Bollondi, L., Thompson, N. & Hayes., B. (2002), 'Automated recognition of intracellular organelles in confocal microscope images', *Traffic* **3**(1), 66.

Fink, J., Aturaliya, R., Davis, M., Zhang, F., Hanson, K., Teasdale, M. & Teasdale, R. (2006), 'Locate: A protein subcellular localization database', *Nucl. Acids Res.* **34**((database issue)).

Haralick, R. (1979), 'Statistical and structural approaches to texture', *Proceedings of the IEEE* **67**(5), 768–804.

Haralick, R., Shanmugam, K. & Dinstein, I. (1973), 'Textural features for image classification', *IEEE Trans. On SMC* **SMC-3**(6), 610 – 621.

Huang, K. & Murphy, R. (2004), 'From quantitative microscopy to automated image understanding', *J. Biomed. Opt.* **9**(5), 893–912.

Khotanzad, A. & Hong, Y. H. (1990), 'Invariant image recognition by zernike moments', *IEEE Trans. Pattern Anal. Mach. Intell.* **12**(5), 489–497.

Lang, P., Yeow, K., Nichols, A. & Scheer, A. (2006), 'Cellular imaging in drug discovery', *Nature Reviews Drug Discovery 5* **5**, 343–356.

Matthiessen, M. (2003), 'Biowaredb: the biomedical software and database search engine', *Bioinformatics* **19**(17), 2319.

Murphy, R., Velliste, M. & Porreca, G. (2003), 'Robust numerical features for description and classification of subcellular location patterns in fluorescence microscope images', *J. VSLI Sig. Proc.* **35**, 311–321.

Pham, T. D., Crane, D. I., Tran, T. H. & Nguyen, T. H. (2004), 'Extraction of uorescent cell puncta by adaptive fuzzy segmentation', *Bioinformatics* **20**(14), 2189–2196.

Pham, T., Tran, D., Zhou, X. & Wong, S. (2006), 'Integrated algorithms for image analysis and identification of nuclear division for high-content cell-cycle screening', *Int. J. Computational Intelligence and Applications* **6**(1), 21–43.

Roques, E. & Murphy, R. (2002), 'Objective evaluation of differences in protein subcellular localisation', *Traffic* **3**, 61–65.

Stow, J. & Teasdale, R. (2005), *Encyclopedia of Genetics, Genomics, Proteomics and Bioinformatics*, John Wiley and Sons, chapter Expression and localization of proteins in mammalian cells.

Thiran, J.-P., Macq, B. & Mairesse, J. (1994), Morphological classification of cancerous cells, *in* 'Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference', Vol. 3, pp. 706–710.

Yu, C., Chen, Y., Lu, C. & Hwang, J. (2006), 'Prediction of protein subcellular localization', *Proteins* **Epub ahead of print**.

Zernike, F. (1934), *Physica* **1**(689).

Quantification of Neural Images using Grey Difference

Donggang Yu^{1,2} , Tuan D. Pham^{1,2} Hong Yan^{3,4} and Denis I Crane^{5,6}

^{1}School of Mathematics, Physics and Information Technology,
^{2} Bionformatics Applications Research Centre,
James Cook University, Townsville, QLD 4811, Australia
^{3}Department of Electronic Engineering,
City University of Hong Kong, Kowloon, Hong Kong
^{4} School of Electrical and Information Engineering,
University of Sydney, NSW 2006, Australia
^{5}School of Biomolecular and Biomedical Science,
^{6} Eskitis Institute for Cell and Molecular Therapies,
Griffith University, Nathan, Qld 4111, Australia

Abstract

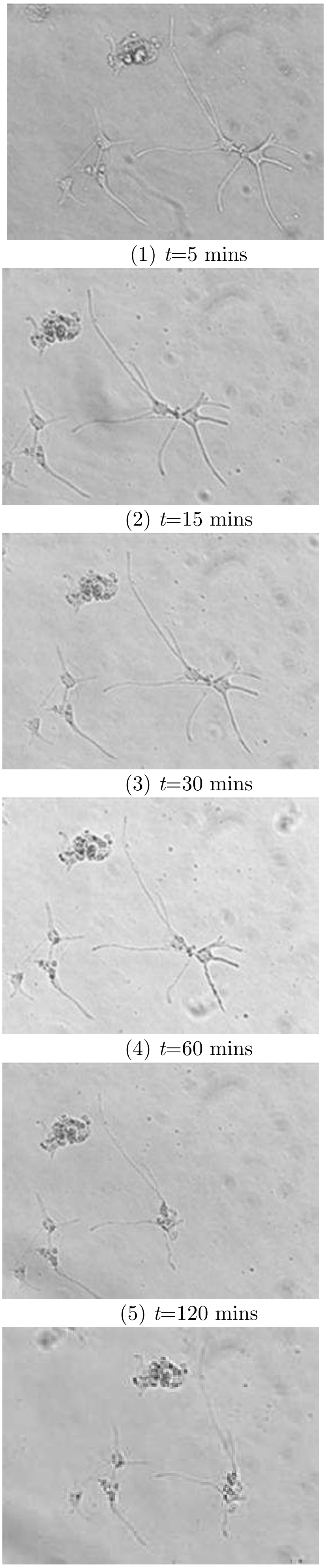
We present new algorithms for segmenting neuron images which are taken from cells being grown in culture with oxidative agents. Information from changing images can be used to compare changes in neurons from the Zellweger mice to those from normal mice. Image segmentation is the first and major step for the study of these different types of processes in neuron cells. It is difficult to segment it as these neuron cell images from stained fields and unimodal histograms. In this paper we develop an innovative strategy for the segmentation of neuronal cell images which are subjected to stains and whose histograms are unimodal. The proposed method is based on logical analysis of grey difference. Two key parameters, window width and logical threshold, are automatically extracted to be used in logical thresholding method. Spurious regions are detected and removed by using hierarchical filtering window. Experiment and comparison results show the efficiency of our algorithms.

Keywords: Neuron cell imaging, segmentation, grey difference, distance difference, filtering window.

1 Introduction

Information taken from images of neuron cells being grown in culture with oxidative agents allows life science researchers to compare changes in neurons from the Zellweger mice to those from normal mice. Neuron degeneration refers to the excessive damage or loss of neurons, or brain and spinal cord cells which perform different functions such as controlling movement, processing sensory information, and making decisions. Neuron degenerative diseases can cause devastating effects on an individual. It is clear that image analysis and recognition are useful tools to help our study of the neuron degeneration in a human disorder called Zellweger syndrome. In our study, the cells are from mice that are a model of the Zellweger syndrome, a severe neuron degenerative disorder characterised by death in the first 16 months after birth, severe dysmorphia, hypotonia, and other widespread tissue defects. This disorder arises because of defects in cellular organelles called peroxisomes, that are required for a number of essential cellular metabolic functions. We have hypothesised that the loss of peroxisomes in neurons results in these cells being susceptible to oxidative stress, because peroxisomes contain a number of important antioxidant enzymes, including catalase needed to break down hydrogen peroxide that is made in cells. In response to oxidative stress, we propose that these neurons will deteriorate. In morphological terms, we expect to see this initial deterioration as the contraction, and eventually loss, of processes of neurons grown in culture. Given the above motivation, image analysis by segmentation is an efficient method that allows us to measure the changes in cell process number and length from images taken of cells being challenged in culture with oxidative agents. The changes in neurons from the Zellweger mice can be compared to those from normal mice in a quantitative manner based on image analysis and recognition. Some neuron cell images subjected to 350 $\mu M\mathrm{H}_2O_2$ with different time (t=5, 15, 30, 60, 120 and 180 mins) are shown in Fig. 1. So the first and most important step is extract neuron images from original images which have stained fields and unimodal histograms. Many methods are investigated for image segmentation. Application of each approach can be useful for solving some particular problem. However, in general, segmentation of nontrival images is still one of the most difficult task in image processing.

In order to segment object images from poor quality images with shadows, nonuniform illumination, low contrast, large signal dependent noise, smear and smudge, it is essential to threshold the image reliably. Therefore, thresholding an intensity image into two levels is the first step and also a critical part in most image analysis systems as any error in this stage will propagate to all later processing, analysis, recognition etc. Although many thresholding techniques, such as global (Ostu 1978) (Lee, Chung & Park 1990) (Pham & Crane 2005) (Chi, Yan & Pham 1996) and local thresholding (Deravi & Pal 1983) (Nakagawa & Rosenfeld 1979) algorithms, multi thresholding methods (Papamarkos & Gatos 1994) and unimodal threshholding (Rosin 2001) have been developed in the past, it is still difficult to deal with images with very low quality. Major problems of segmenting poor quality images are variable background intensity due to nonuniform illumination, low local contrast due to smear or smudge and shadows. For example, one group of screening neuron cell images from stained fields and monomodal histograms are shown in Fig. 1. In this paper, we propose innovative segmentation algorithms of neuron cell images, which are a thresholding method based on logical level technique with difference analysis of the grey region and filtering window with contained condition. This thresholding method is used to binarize poor quality greyscale neuron cell images. The thresholding parameters of this method are automatically selected based on difference analysis of grey region. In order to segment neuron cell images, binarized images



(1) t=5 mins

(2) t=15 mins

(3) t=30 mins

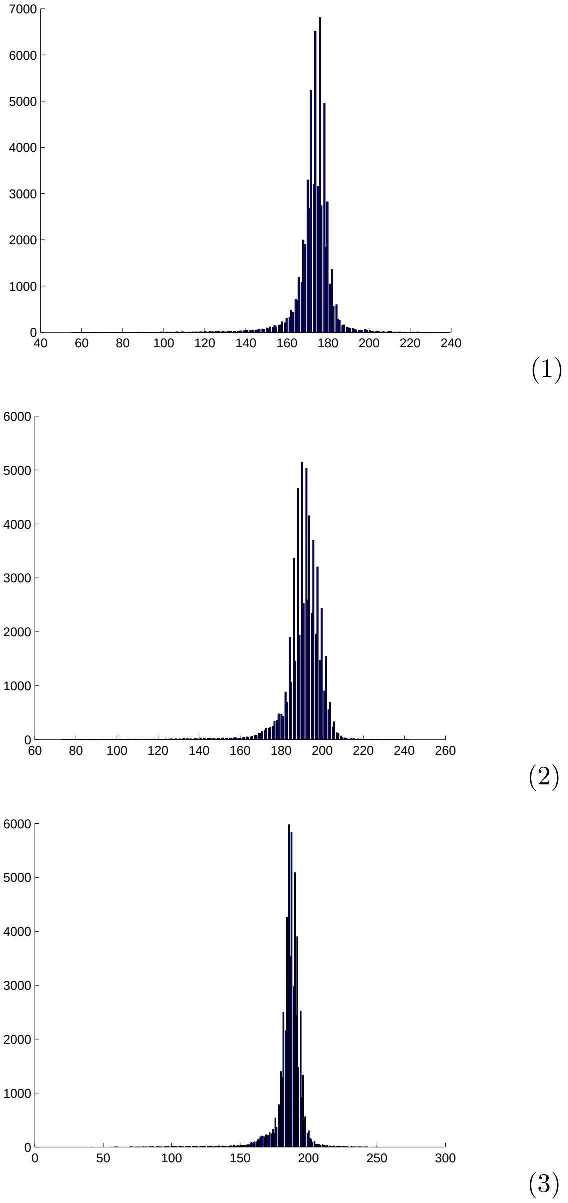
(4) t=60 mins

(5) t=120 mins

(6) t=180 mins

Figure 1: One example of neuron cell screening.

are filtered by hierarchical filtering window with contained condition. Our method can deal with variable background intensity caused by nonuniform illumination, shadow, smear or smudge and low contrast without obvious loss of useful information. This paper is orgnized as follows. In Section 2, we briefly review related works on image thresholding techniques. In Section 3, segmentation algorithms of neuron cell images, logical level technique with difference analysis of grey region and filtering window with contained condition are described. In Section 4, we illustrate the performance of the proposed methods using several experiments, and compare several experiments of the proposed method to some related segmentation methods. We then conclude our analyzer in the final section.



(1)

(2)

(3)

Figure 2: Unimodal histogram of images in Figs. 1(1,5,6).

2 Related work

The most commonly used global thresholding techniques are based on histogram analysis (Ostu 1978)(Lee et al. 1990). Threshold is determined from the measure that best separates the levels corresponding to the peaks of the histogram, each of which corresponds to image pixels of a different part like background or objects in the image. A threshold is an intensity value which is used as the boundary between two classes of a binary segmented image. These methods do not work well for the poor quality images with shadows, inhomogeneous backgrounds, complex background patterns which may have a histograms that contains a single peak. For example, the histograms of images in Figs. 1(1,5,6) are single peak (unimodal), which are shown in Fig. 2. In this case, a single threshold could not result in an accurate binary image. For example, if the neuron cell images in Figs. 1(1-6) are segmented by Otsu's method, then binarization results are shown in Fig. 3.

Distinct from thresholding method, k-Means clustering is involved to determine classes themselves, rather than a threshold value (Zhang 2000). Fuzzy c-Means Clustering is used to segment images, which is called as FCM (Chi et al. 1996). However these techniques use only intensity data of images to perform segmentations, and as the spatial structure of the images is not taken into account. Therefore, the segmentation results are similar to those by Otsu's method or more not efficient.

The segmentation result by Sobel edge method (Gonzalez, Woods & Pham 2002)is not suitable for the extraction of neuron images because only edge information of objects is extracted and some useful parts of neuron image are mist (see Fig. 4). Other segmentation methods, background subtraction methods cannot be used to extract neuron images for the database of neuron images used here because background subtraction methods should be used in no more changed background for extracting changed objects (such as moving objects)(Gonzalez et al. 2002).

One unimodal thresholding is approached by (Rosin 2001). The threshold point is selected as the histogram index value that maximises the perpendicular distance between the straight line (drawing from the peak to the high end of the histogram) and histogram line. However, this method relies on several assumptions and was unable to accurately segment the neuron cell images in Figs. 1(1-6), and their segmentation results are shown in Fig. 5.

3 Logical level technique with difference analysis of grey region and filtering window with contained condition

3.1 Logical level technique

Logical level technique are developed to be used to segment document images images by Kamel, Zhao (Kamel & Zhao 1993), Yang and Yan (Yang & Yan 2000). After analyzing integrated function algorithm (Trier & Taxt 1995), Kamel and Zhao proposed Logical level technique. The basic idea is comparing the grey level of the processed pixel or its smoothed grey level with some local averages in the neighborhoods, and the comparison results are regarded as derivatives. Therefore, pixel labeling, detection and extraction using the derivatives, the logical bound on the ordered sequences and the window width range can be adopted. This technique processes each pixel by simultaneously comparing its grey level or its smoothed grey level with four local averages in the selected window region. Suppose selected window is "W". The window region is $(2W+1)^2$. Let the start point of the image be upper-left and $f(i,j)$ be grey intensity of coordinates (i,j), and it is eight neighboring.

Suppose each neighbour point (x,y) is the center of region $(2W+1)^2$, then the average grey intensity

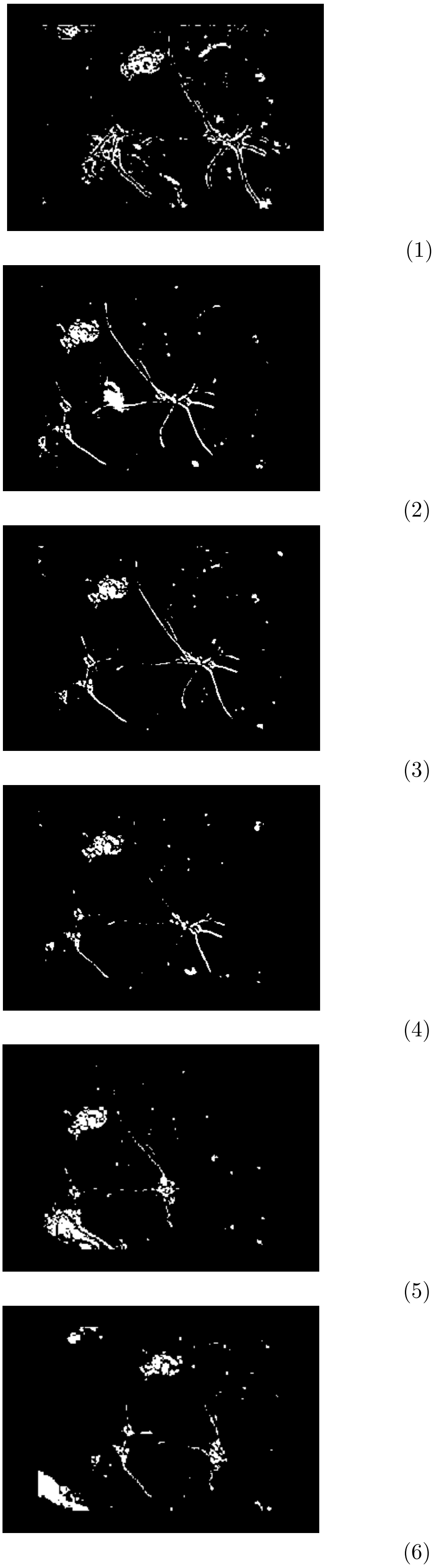
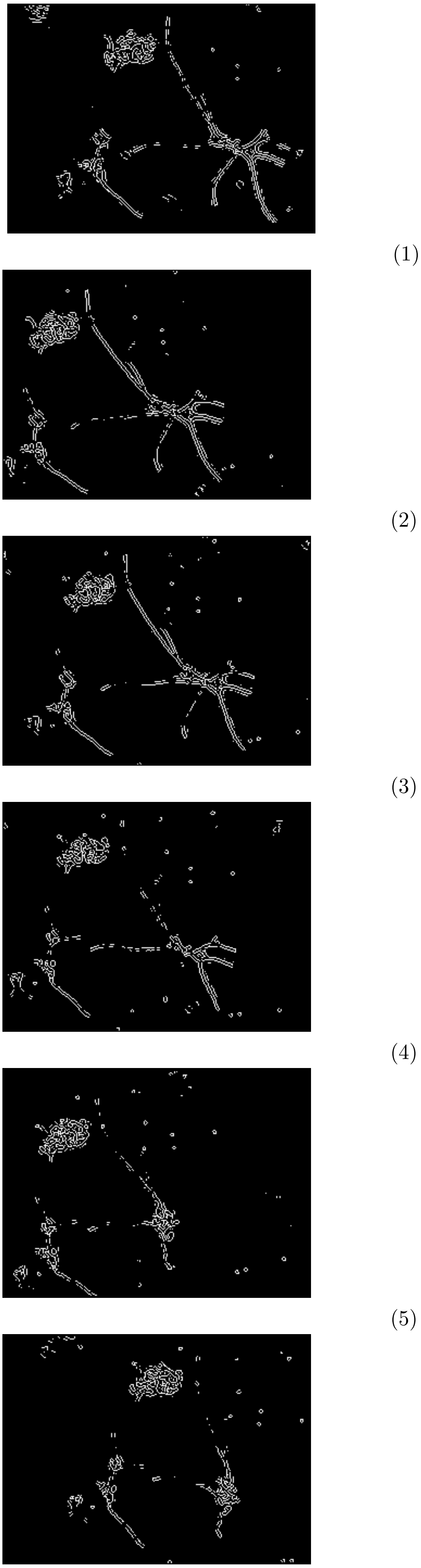


Figure 3: Segmentation results of images in Figs. 1(1-6) by Otsu's method.



(1)

(2)

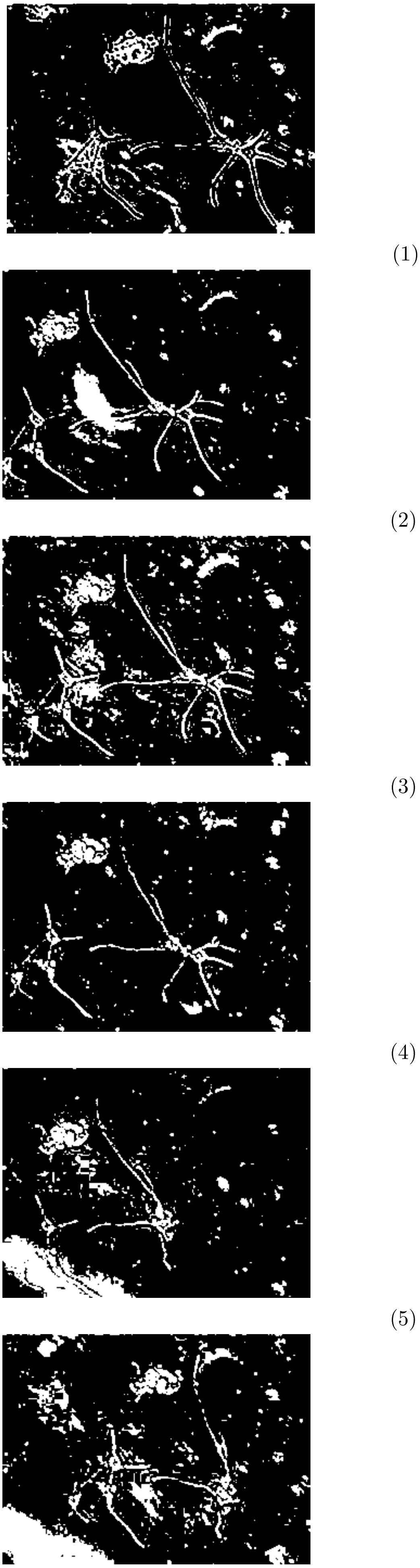
(3)

(4)

(5)

(6)

Figure 4: Segmentation results of images in Figs. 1(1-6) by Sobel edge method.



(1)

(2)

(3)

(4)

(5)

(6)

Figure 5: Segmentation results of images in Figs. 1(1-6) by unimodal thresholding method.

$lp(k)$ of region $(2W+1)^2$ is

$$lp(k) = \frac{\sum_{-W \le m \le W} \sum_{-W \le n \le W} f(x+m, y+n)}{(2W+1)^2} \tag{1}$$

where if k=0, $x=i$ and $y=j+1$; k=1, $x=i-1$ and $y=j+1$; k=2, $x=i-1$ and $y=j$; k=3, $x=i-1$ and $y=j-1$; k=4, $x=i$ and $y=j-1$; k=5, $x=i+1$ and $y=j-1$; k=6, $x=i+1$ and $y=j$; k=7, $x=i+1$ and $y=j+1$. Therefore grey region difference ($llp(k)$) between $lp(k)$ and $f(i,j)$ can be found

$$llp(k) = lp(k) - f(i,j) \ge T \quad k = 0, 1, ..., 7 \tag{2}$$

where "T" is predetermined parameter.
The logical level technique is

$$\text{b(i,j)=}\begin{cases} 1 & \text{if } \{(llp(0) \bigwedge llp(4)) \bigvee \\ & (llp(2) \bigwedge llp(6)) \bigvee \\ & (llp(1) \bigwedge llp(5)) \bigvee \\ & (llp(3) \bigwedge llp(7))\ \} \quad \text{is true} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where "1" represents object and "0" to represent background in the resulting binary image.

3.2 Innovative logical logical level technique with difference analysis of grey region and filtering window with contained condition

We can find that logical level technique need two key parameters, window "W" and threshold "T". However, predetermination of these parameters is difficult, and no efficient method can be found from logical level technique, hence we have developed a new method to determine parameters, window "W" and threshold "T" automatically based on the analysis grey regions. Grey region can be defined as the region between each pair of neighbour grey peak and valley points in horizontal and vertical direction. Mathematically, the peak and valley points of image grey histogram are the points which make the first order derivative of image grey function equal to zero. For each row the peak point set, P_h, can be found as follows.
If the point is starting point of a row and each row has *col* points, the peak points in horizontal direction can be found as follows:
(1) if $f(i,1) > f(i,2)$, then the point $P_h(i,1)$ is a peak point.
(2) if $f(i,1), ..., f(i,n) > f(i,n+1)$ and $f(i,1) = ... = f(i,n)$ $(n > 1)$, then the point $P_h(i,1)$ is a peak point.
If point is last point of a row, the peak points in horizontal direction based on two cases:
(3) if $f(i, col-1) < f(i, col)$, then the point $P_h(i, col)$ is a peak point.
(4) if $f(i, col-n-1) < f(i, col)$ and $f(i, col-1) = ... = f(i, col-n)$ $(n > 1)$, then the point $P_h(i, col)$ is a peak point.

In other cases, the peak points in horizontal direction based on two cases:
(5) if $f(i, j-1) < f(i,j)$ and $f(i,j) > f(i, j+1)$, then the point $P_h(i,j)$ is a peak point.
(6) if $f(i, j-1) < f(i,j), ..., f(i, j+n)$ and $f(i,j) ... f(i, j+n) > f(i, j+n+1)$ where $f(i,j) = ... = f(i, j+n)$ $(n > 1)$, then the point $P_h(i,j)$ is a peak point.
Similarly, the valley point set, V_h can be found for each row of image. Similarly, the peak and valley point sets, P_v and V_v, of each collum can be found. Grey regions can be calculated based on found peak and valley point sets, P_h, V_h, P_v and V_v. For each grey region two parameters are calculated. The first parameter is the grey difference between each pair of neighbour peak and valley points, which can be represented as $H_g(m), m = 1, 2, ...k$, where k is region number for all rows of an image. The second parameter is distance difference between each pair of peak and valley points, which can be represented as $H_d(m), m = 1, 2, ...k$. Furthermore, one new data set of grey region in which the number of points that have same grey difference and distance difference is found. It can be represented with $H_{dg}(m), m = 1, 2, ...kn$ (k_n being the number of groups). Sort $H_{dg}(m), m = 1, 2, ...kn$ based on $H_{dg}(m)$ get a decreasing data set, $H_{dgd}(m), m = 1, 2, ...kn$. Therefore $H_{dgd}(0)$ is the first number of grey regions with same grey and distance difference, and it is largest. If first tk groups are summed

$$S_{tk} = \sum_{m=1}^{tk} H_{dgd}(m). \tag{4}$$

Parameter tk is selected to meet $(S_{tk}/k) \ge 0.7$, where k is region number for all rows of an image. For example, for the image in Fig. 1(6) tk=81, kn=931, (S_{tk}=8397 and k=11961. We can see only 81 groups of grey region contain 8397 grey regions which is approximately equal to 70% of k=11961. Therefore, here 81 groups of grey region represent major property of region distribution of the image in horizontal direction. Based on this idea, window parameter "W" and threshold "T" can be determined. "W" and "T" are selected as mean region distance and region grey difference of tk groups of grey region respectively. That is

$$W_h = \frac{\sum_{m=1}^{tk} H_{dd}(m)}{tk}. \tag{5}$$

$$T_h = \frac{\sum_{m=1}^{tk} H_{gd}(m)}{tk}. \tag{6}$$

where $H_{dd}(m)$ and $H_{gd}(m)$ is region distance and grey difference of each group of tk groups in horizontal direction respectively.
Similarly, the peak and valley points, related analysis parameters, W_v (window parameter in vertical direction) and T_v (thresholding parameter in vertical direction) in vertical direction of images can be found. The final window parameter is $W = (W_h + W_v)/2$, and thresholding parameter is $T = (T_h + T_v)/2$. We can find all window parameters, "W", and thresholds, "T", for the images in Fig. 1 based the above algorithm. For example, $W = 5$ and $T = 6.95$ for the image in Fig. 1(1), and $W = 4$ and $T = 6.5$ for the image in Fig. 1(6). Based on the found parameters, "W", "T" and logical thresholding algorithm, the images in Fig. 1 can be extracted, and shown in Fig. 6.

3.3 Filtering window with contained condition

As average smoothed grey and grey difference information of image window is used, the algorithm can binarized poor quality greyscale image which has variable background intensity, smear or smudge and low contrast. However, if a region meets the conditions of binarization, it can be selected as an object image. For example, some spurious regions, which are isolated with small size, are made and shown in Fig. 6. In many scientific applications, objects of interest (such as the neurons in Fig. 1) are large in size and

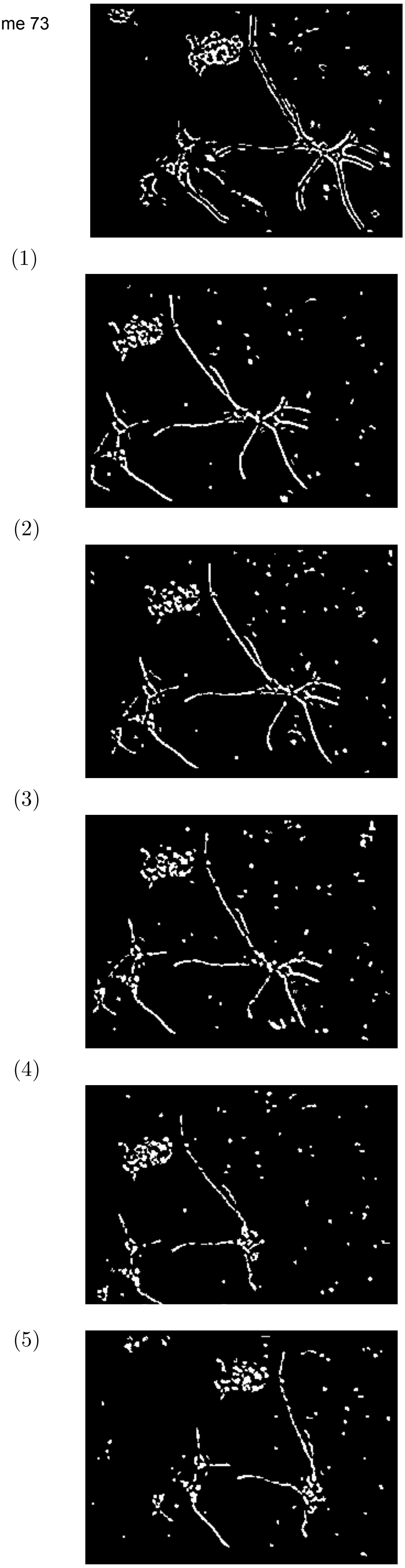


Figure 6: Segmentation results of cell images in Figs. 1-6 by innovative logical thresholding.

close together. Spurious regions on other hand are usually small and isolated by comparison. For example, neuron images belong to such a case. Therefore, we can detect and remove some spurious region based above idea.

The algorithm can be described as follows:

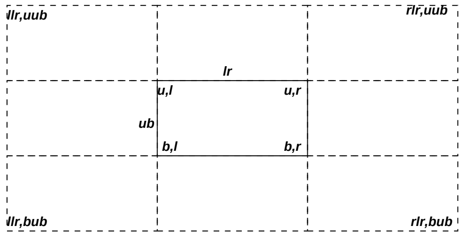


Figure 7: Filtering window with one object region and detection region (filtering window plus its eight neighbouring regions of equal size).

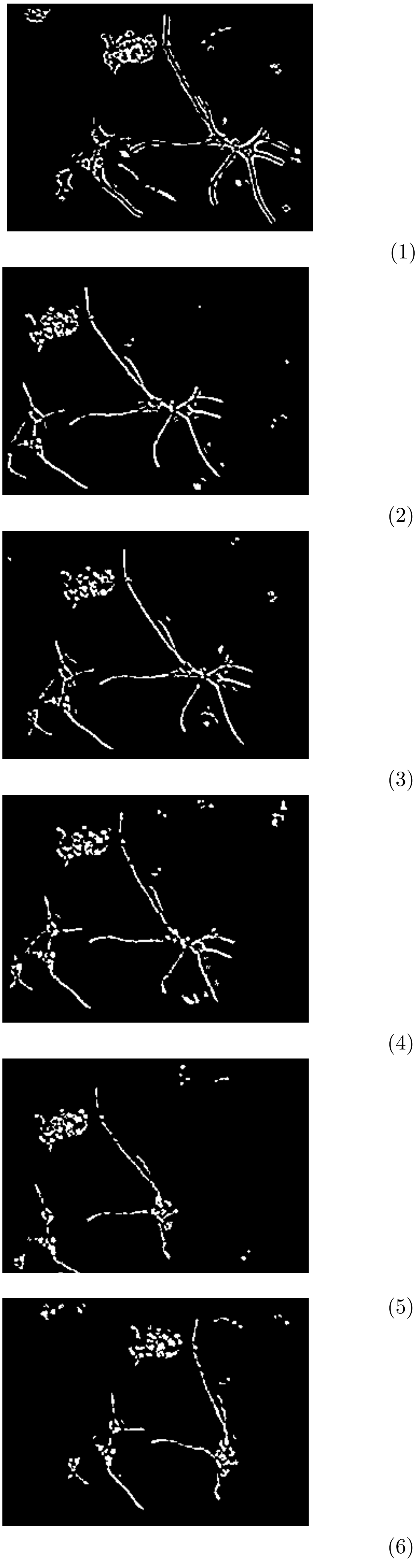
(1) Find all regions of binarization image, which can be represented as $R(k), k = 1, ...rn$, where rn is the number of regions and $R(k)$ is k-th region's area size.
(2) Sort $R(k), k = 1, ...rn$ based their area size in increasing order, and it is represented as $SR(k), k = 1, ...rn$.
(3) For each region (starting with smallest):
(3.1) Find the minimum bounding rectangle which covers the region (this is called filtering window).
(3.2) Determine the detection region R_d. R_d will consist of the filtering window plus it's eight neighbouring regions of equal size. The detection region is shown in Fig. 7, where $(l, u), (r, u), (l, b)$ and (r, b) are the coordinates of four corners of found minimum rectangle respectively, $lr = r - l$ and $ub = b - u$ are sizes of the rectangle respectively, and $llr = l - lr$, $rlr = r + lr$, $uub = u - ub$ and $bub = b + ub$.
(3.3) Detect whether there is a point in another object region in the detection region R_d. If so, then remove the processed region from $R(k)$ if not, then keep the processed region.
Based on the above algorithm, the images in Fig. 6 can be processed and shown in Fig. 8.

In above algorithm, filtering window only contains one object region. Hierarchical processing can be done based on new filtering window which consists of two neighbour object regions. The processing method can be described similar to above algorithm. Based on the above procedure, the images in Fig. 8 can be processed and shown in Fig. 9.

Furthermore, filtering window with big size can be selected. However, we should consider whether selected filtering window is reasonable based on the prior knowledge and result of binarization image. Such contained conditions should be considered to select the size of filtering window. For example, if the size of valuable object image is little and the number of object image is not large enough, little size of filtering window should be selected. For our processed neural images, only two sizes of filtering window are used, the first filtering window with one object region and the second filtering window with two object regions.

4 Experimental results

We have tested some neuron cell images which taken from screenings of neuron cell images based on our innovative algorithm. For example, six neuron cell images subjected to 350 $\mu M H_2 O_2$ are shown in Fig. 1 which are taken in different time (t=5, 15, 30, 60, 120 and 180 mins respectively). We can see that the poor quality images of neuron cells are with shadows, in-



(1)

(2)

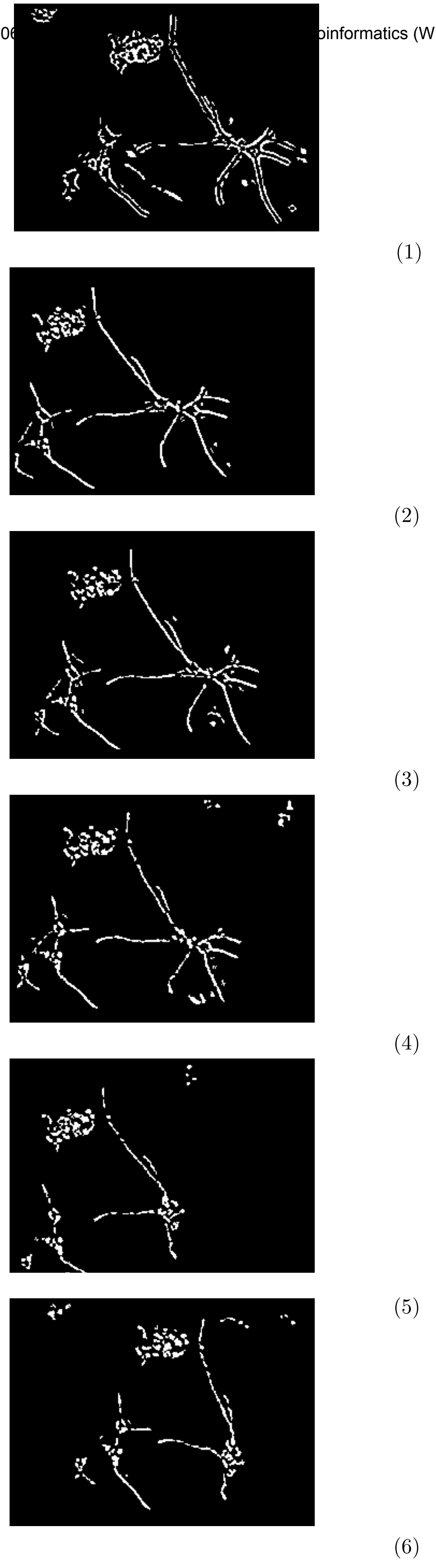
(3)

(4)

(5)

(6)

Figure 8: The processed results of segmentation results in Fig. 6 by filtering window with one object region.



(1)

(2)

(3)

(4)

(5)

(6)

Figure 9: The processed results of segmentation results in Fig. 8 by filtering window with two object regions.

homogeneous backgrounds, complex background patterns which may have a histograms that contains a single peak. Here, histograms of six neuron cell images are single peak. Based on our method, the neuron cell images are processed by innovative logical thresholding with grey difference analysis, and extracted neuron cell images are shown in Fig. 6 firstly. We can see that cellular organelles are well extracted, but there are some spurious regions in Fig. 6. This is because our method is thresholding based on grey difference analysis. If thresholding conditions ("W" and "T") are meet for some regions which are not belong to neuron cell regions, the spurious regions are formed. However, most spurious regions are isolated because of thresholding with grey difference. Therefore, extracted images in Fig. 6 can further be processed by using hierarchical filtering window secondly. Here two sizes of filtering windows are used to process the neuron cell images in Fig. 6, and the extracted neuron cell images are shown in Fig. 8 and Fig. 9 respectively. We can see that cellular organelles are extracted, and some spurious regions are removed.

For some neuron cell images in Figs. 1(1-6), Ostu's method (Ostu 1978) and unimodel thresholding (Rosin 2001) are used to extract neuron cell images, and the processed results are shown in Figs. 3, 5 respectively. Threshold of all three methods (Ostu 1978) and (Rosin 2001) is determined from the measure that best separates the levels corresponding to the peaks of the histogram, each of which corresponds to image pixels of a different part like background or objects in the image. The threshold is an intensity value which is used as the boundary between two classes of a binary segmented image. If there are inhomogeneous backgrounds and complex background patterns in neuron cell images such as Figs. 1(1-6), shadows may become object, and some parts of objects may become background, which are caused by nonuniform illumination, shadow, smear or smudge and low contrast. In these cases no good segmentation results can been got by the intensity thresholding. It is clear our algorithm is more efficient by comparing the result of proposed method (see Fig. 9) with those of other three methods (see Figs. 3 and 5). Also, the segmentation result by Sobel edge method is not suitable because only edge information of objects is extracted and some useful parts of neuron image are mist (see Fig. 4).

5 Conclusion

In this paper, we have developed a segmentation algorithm for screening neuronal cell images based on logical thresholding of grey difference. Thresholding parameter can be selected automatically based on analysis of grey difference region. Our method can effectively segment grey scale images such as screening neuron cell images which have variable background intensity caused by nonuniform illumination, shadow, smear or smudge and low contrast. Proposed filtering window can be used to remove spurious regions of binarization images of neuron cell images. Experiment and comparison results show the efficiency of our algorithms.

Acknowledgement- This work is supported by ARC-DP grant DP0665598.

References

Chi, Z., Yan, H. & Pham, T. 1996 Fuzzy Algorithm: With Application to Image Processing and Pattern Recognition, World Scientific Publishing Co., Singapore.

Deravi, F. & Pal, S.K. 1983Grey level thresholding using second order statistics, Pattern Recognition Lett, **1**, pp. 417–422.

Gonzalez, R.C. & Woods, R.E. 2002 Digital Image Processing, 2nd edition, New Jersey, Prentice Hall.

Kamel, M. & Zhao, A. 1993 Extraction of binary character/ graphics images from greyscale document images, CVGIP: Graphical Models Image Process. **55**, pp. 203–217.

Lee, S.U., Chung, S.Y. & Park, R.H. 1990, A comparative perfor mance study of several global thresholding techniques for segmentation, CVGIP, **52**, pp 171–190.

Nakagawa, Y. & Rosenfeld, A. 1979Some experiments on variable thresholding, Pattern Recognition, **11**, pp. 191–204.

Ostu, N. 1978, A thresholding selection method from greylevel histogram, IEEE Trans. Systems Man Cybernet, **SMC8**, pp. 62–66.

Papamarkos, N. & Gatos, B. 1994 A new approach for multilevel threshold selection, CVGIP: Graphical Models Image Process, **56**, pp. 357–370.

Pham, T.D. & Crane, D.I. 2005 Segmentation of neuronal-cell images from stained fields and monomodal histograms, Proc. 27th Annual Int. Conf. IEEE Engineering in Medicine and Biology Society, **3.5**, pp. 7–13.

Rosin, P.L. 2001 Unimodal thresholding, Pattern Recognition, **34**, pp. 2083–2096.

Trier, O.D. & Taxt, T. 1995 Improvement of intergrated function algorithm' for binarization of document images, Pattern Recognition Lett, **16**, pp. 277–283.

Yang, Y. & Yan H. 2000 An adaptive logical method for binarization of degraded document images. Pattern Recognition **33(5)**, pp. 787–807.

Zhang, B. 2000 Generalized k-harmonic means-boosting unsupervised learning, Technical Reprt HPL-2000-137, Hewlett-Packard Labs..

Author Index

Ahsan, Nasir, 3
Azok, Joseph, 61

Bagirov, Adil M., 23
Bailey, Timothy L., iii
Bain, Michael, 3
Bean, R.W., 29
Beck, Dominik, 61
Bodén, Mikael, iii, 39, 49, 55
Brandl, Miriam, 61
Brennan, Marie-Luise, 61

Chakka, Nagesh, 15
Crane, Denis I., 73

Daggard, Grant, 35
Davis, Lynne, 39
Dawes, Ian, 3

Fink, J.L., 67

Gaëta, Bruno, 3
Gallagher, Marcus, 49
Gready, Jill E., 15

Hamilton, N., 67
Hanson, K., 67
Hawkins, John, 39, 55
Hazen, Stanley L., 61
Hoehn, Gerald, 61
Hogan, James M., 9
Hu, Hong, 35

Karunaratne, S., 67

Li, Jiuyong, 35
Li, King, 61

Maetschke, Stefan R., 9, 39, 49
Mardaneh, Karim, 23
McLachlan, G.J., 29

Ng, S.-W., 29
Ng, S.K., 29

Pantelic, R., 67
Pham, Tuan D., 61, 73
Potter, John, 3

Shi, Mingren, 35

Teasdale, R.D., 67
Temple, Mark, 3
Towsey, Michael W., 9

Wang, Honghui, 61
Wang, Hua, 35
Wong, Stephen T.C., 61

Yan, Hong, 73
Yu, Donggang, 73

Zhou, Xiaobo, 61

Recent Volumes in the CRPIT Series

ISSN 1445-1336

Listed below are some of the latest volumes published in the ACS Series *Conferences in Research and Practice in Information Technology*. The full text of most papers (in either PDF or Postscript format) is available at the series website `http://crpit.com`.

Volume 53 - Conceptual Modelling 2006

Edited by Markus Stumptner, *University of South Australia*, Sven Hartmann, *Massey University, New Zealand* and Yasushi Kiyoki, *Keio University, Japan*. January, 2006. 1-920-68235-X.

Contains the proceedings of the Third Asia-Pacific Conference on Conceptual Modelling (APCCM2006), Hobart, Tasmania, Australia, January 2006.

Volume 54 - ACSW Frontiers 2006

Edited by Rajkumar Buyya, *University of Melbourne*, Tianchi Ma, *University of Melbourne*, Rei Safavi-Naini, *University of Wollongong*, Chris Steketee, *University of South Australia* and Willy Susilo, *University of Wollongong*. January, 2006. 1-920-68236-8.

Contains the proceedings of the Fourth Australasian Symposium on Grid Computing and e-Research (AusGrid 2006) and the Fourth Australasian Information Security Workshop (Network Security) (AISW 2006), Hobart, Tasmania, Australia, January 2006.

Volume 55 - Safety Critical Systems and Software 2005

Edited by Tony Cant, *University of Queensland*. April, 2006. 1-920-68237-6.

Contains the proceedings of the 10th Australian Workshop on Safety Related Programmable Systems, August 2005, Sydney, Australia.

Volume 56 - Vision in Human-Computer Interaction

Edited by Roland Goecke, Antonio Robles-Kelly, and Terry Caelli, *NICTA*. November, 2006. 1-920-68238-4.

Contains the proceedings of the HCSNet Workshop on the Use of Vision in Human-Computer Interaction (VisHCI 2006).

Volume 57 - Multimodal User Interaction 2005

Edited by Fang Chen and Julien Epps *National ICT Australia*. April, 2006. 1-920-68239-2.

Contains the proceedings of the NICTA-HCSNet Multimodal User Interaction Workshop 2005, Sydney, Australia, 13-14 September 2005.

Volume 58 - Advances in Ontologies 2005

Edited by Thomas Meyer, *National ICT Australia, Sydney* and Mehmet Orgun *Macquarie University*. December, 2005. 1-920-68240-6.

Contains the proceedings of the Australasian Ontology Workshop (AOW 2005), Sydney, Australia, 6 December 2005.

Volume 60 - Information Visualisation 2006

Edited by Kazuo Misue, Kozo Sugiyama and Jiro Tanaka. February, 2006. 1-920-68241-4.

Contains the proceedings of the Asia-Pacific Symposium on Information Visualization (APVIS 2006), Tokyo, Japan, February 2006.

Volume 61 - Data Mining 2006

Edited by Simeon Simoff, *University of Technology, Sydney* and Graham Williams *Australian Taxation Office and University of Canberra*. December, 2006. 1-920-68242-2.

Contains the proceedings of the Australasian Data Mining Conference (AusDM 2006), December 2006.

Volume 62 - Computer Science 2007

Edited by Gillian Dobbie, *University of Auckland, New Zealand*. January, 2007. 1-920-68243-0.

Contains the proceedings of the Thirtieth Australasian Computer Science Conference (ACSC2007), Ballarat, Victoria, Australia, January 2007.

Volume 63 - Database Technologies 2007

Edited by James Bailey, *University of Melbourne* and Alan Fekete, *University of Sydney*. January, 2007. 1-920-68244-9.

Contains the proceedings of the Eighteenth Australasian Database Conference (ADC2007), Ballarat, Victoria, Australia, January 2007.

Volume 64 - User Interfaces 2007

Edited by Wayne Piekarski, *University of South Australia*. January, 2007. 1-920-68245-7.

Contains the proceedings of the Eighth Australasian User Interface Conference (AUIC2007), Ballarat, Victoria, Australia, January 2007.

Volume 65 - Theory of Computing 2007

Edited by Joachim Gudmundsson, *NICTA, Australia* and Barry Jay *UTS, Australia* . January, 2007. 1-920-68246-5.

Contains the proceedings of the Thirteenth Computing: The Australasian Theory Symposium (CATS2007), Ballarat, Victoria, Australia, January 2007.

Volume 66 - Computing Education 2007

Edited by Samuel Mann, *Otago Polytechnic* and Simon *Newcastle University*. January, 2007. 1-920-68247-3.

Contains the proceedings of the Ninth Australasian Computing Education Conference (ACE2007), Ballarat, Victoria, Australia, January 2007.

Volume 67 - Conceptual Modelling 2007

Edited by John F. Roddick, *Flinders University* and Annika Hinze, *University of Waikato, New Zealand*. January, 2007. 1-920-68248-1.

Contains the proceedings of the Fourth Asia-Pacific Conference on Conceptual Modelling (APCCM2007), Ballarat, Victoria, Australia, January 2007.

Volume 68 - ACSW Frontiers 2007

Edited by Ljiljana Brankovic, *University of Newcastle*, Paul Coddington, *University of Adelaide*, John F. Roddick, *Flinders University*, Chris Steketee, *University of South Australia*, Jim Warren, *the University of Auckland*, and Andrew Wendelborn, *University of Adelaide*. January, 2006. 1-920-68249-X.

Contains the proceedings of the ACSW Workshops - The Australasian Information Security Workshop: Privacy Enhancing Systems (AISW), the Australasian Symposium on Grid Computing and Research (AUSGRID), and the Australasian Workshop on Health Knowledge Management and Discovery (HKMD), Ballarat, Victoria, Australia, January 2007.

Volume 72 - Advances in Ontologies 2006

Edited by Mehmet Orgun *Macquarie University* and Thomas Meyer, *National ICT Australia, Sydney*. December, 2006. 1-920-68253-8.

Contains the proceedings of the Australasian Ontology Workshop (AOW 2006), Hobart, Australia, December 2006.

Volume 73 - Intelligent Systems for Bioinformatics 2006

Edited by Mikael Boden and Timothy Bailey *University of Queensland*. December, 2006. 1-920-68254-6.

Contains the proceedings of the AI 2006 Workshop on Intelligent Systems for Bioinformatics (WISB-2006), Hobart, Australia, December 2006.