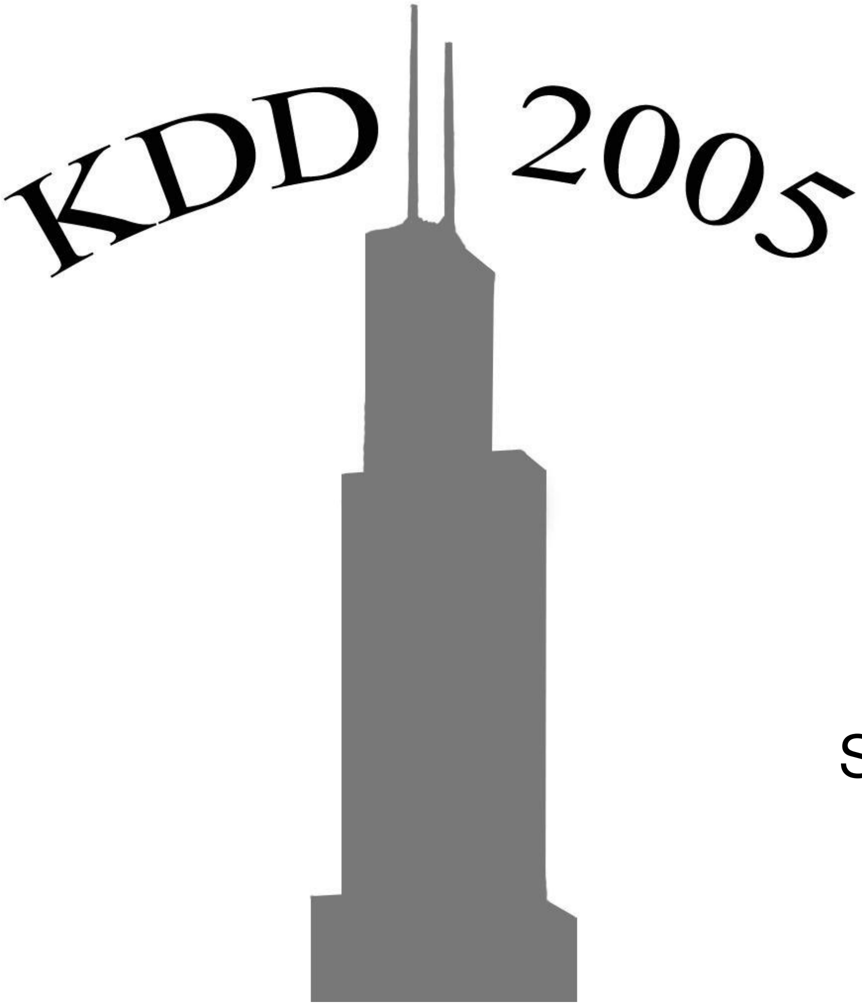
5th International Workshop on Data Mining in Bioinformatics (BIOKDD'05)



Workshop Chairs:

Srinivasan Parthasarathy
Wei Wang
Mohammed Zaki

August 21, 2005
Chicago, Illinois, USA

BIOKDD05: Workshop on Data Mining in Bioinformatics
August 21, 2005
Chicago, IL, USA

in conjunction with
11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining

Srinivasan Parthasarathy
Computer Science and
Engineering Department
Ohio State University
srini@cse.ohio-state.edu
Program Committee Chair

Wei Wang
Computer Science
Department
University of North Carolina
weiwang@cs.unc.edu
Program Committee Co-Chair

Mohammed Zaki
Computer Science
Department
Rensselaer Polytechnic
Institute
zaki@cs.rpi.edu
General Chair

Opening Remarks

Bioinformatics is the science of managing, mining, and interpreting information from biological entities. Genome sequencing projects have contributed to an exponential growth in complete and partial sequence databases. The structural genomics initiative aims to catalog the structure-function information for proteins. Advances in technology such as microarrays have launched the subfield of genomics and proteomics to study the genes, proteins, and the regulatory gene expression circuitry inside the cell. What characterizes the state of the field is the flood of data that exists today or that is anticipated in the future; data that needs to be mined to help unlock the secrets of the cell. Knowledge extracted from such analysis can be used effectively to better design new drugs, offer better medical care via diagnostic tests that combine information from multiple sources, and improve scientific and clinical practice.

While tremendous progress has been made over the years, many of the fundamental problems in bioinformatics, such as protein structure prediction or gene finding, are still open. Data mining will play a fundamental role in understanding gene expression, drug design and other emerging problems in genomics and proteomics. Furthermore, text mining will be fundamental in extracting knowledge from the growing literature in bioinformatics.

The goal of this workshop was to encourage KDD researchers to take on the numerous challenges that Bioinformatics offers. The workshop features an invited talk from a noted expert in the field, and the latest data mining research in bioinformatics from world class researchers. We encouraged papers that propose novel data mining techniques for tasks such as: Gene expression analysis; Protein/RNA structure prediction; Phylogenetics; Sequence and structural motifs; Genomics and Proteomics; Gene finding; Drug design; RNAi and microRNA Analysis; Text mining in bioinformatics; Modeling of biochemical pathways; and Biomedical and clinical informatics.

These proceedings contain 10 papers (5 long and 5 short), out of 20 submissions that were accepted for presentation at the workshop. Each paper was reviewed by at least three members of the program committee. In some cases where there was a wide variance in reviews a fourth was sought. Each long paper selected had at least two strong supporters and no strong detractor. Each short paper selected had at least one strong supporter and typically no strong detractor. As a result along with a distinguished invited talk, we were able to assemble a very exciting program.

We would like to thank all the authors, invited speaker, and attendees for contributing to the success of the workshop. Special thanks are due to the program committee for help in reviewing the submissions.

This workshop follows the previous four highly successful workshops: BIOKDD04, held in Seattle, BIOKDD03, held in Washington, DC; BIOKDD02, held in Edmonton, Canada; and BIOKDD01 held in San Francisco, CA. We expect BIOKDD05 to be equally successful.

International Program Committee

• Raj Acharya (USA) • Srinivas Aluru (USA) • Jean-Francois Boulicaut (France) • Michael Berthold (Germany) • Chris Ding (USA) • Hakan Ferhatosmanoglu (USA) • Vasant Honavar (USA) • Melanie Hilario (Switzerland) • George Karypis (USA) • Stefan Kramer (Germany) • Jayant Haritsa (India) • Stefan Kramer (Germany) • Tung Kum Hoe (Singapore) • Jaewoo Kang (USA) • Mitsunori Ogihara (USA) • Yi Pan (USA) • Isidore Rigoutsos (USA) • Kotagiri Rammohanrao (Australia) • Ambuj Singh (USA) • Hannu Toivonen (Finland) • M. Vidyasagar (India) • Ke Wang (Canada) • Jason T. Wang (USA) • Jiong Yang (USA)

External Reviewers

• Amol Ghoting (USA) • Keith Marsolo (USA) • Matthew Otey (USA) • Duygu Ucar (USA)

Workshop Program & Table of Contents

8:55 9:00am: Opening Remarks

9:00 10:00am: Session I: Proteins

10:00 10:30am: Coffee Break

10:30 12:00: Session II: Genomics and Health

12:00 1:30pm: Lunch

1:30 3:10am: Session III: Sequences and Microarrays

3:10 3:30pm: Coffee Break

3:30 4:30pm: Session IV: Protein Interaction Networks and Bioentity Recognition

4:30 4:45: Closing Comments and Discussion

Motif Discovery for Proteins Using Subsequence Clustering

Hardik A. Sheth
School of Informatics
Indiana University
901 E. 10th St.
Bloomington, IN 47408 3912
hsheth@indiana.edu

Sun Kim
School of Informatics
Center for Genomics and Bioinformatics
Indiana University
901 E. 10th St.
Bloomington, IN 47408 3912
sunkim2@indiana.edu

ABSTRACT

We propose an algorithm for discovering motifs using clustering of subsequences. In our previous approach, we were successful in guiding motif discovery by sampling subsequences and inputting them to an existing motif discovery tool MEME. In this paper, we show that clustering subsequences can also detect motifs without using other motif discovery tools. Generally, motif discovery algorithms do not perform well when the input set consists of non-homogeneous sequences. Clustering tools have the inherent ability to generate clusters of homogeneous sequences when the input sequences are non-homogeneous. For this reason, we use our clustering algorithm to generate aligned subsequence clusters and then rank them according to their information contents to produce final motifs. The algorithm was tested with PROSITE database and the results suggest that the algorithm is very effective in finding motifs even when input sequences are from different protein families.

Categories and Subject Descriptors

I.5.1 [**Pattern Recognition**]: Models - Statistical

General Terms

Algorithms, Performance, Experimentation

Keywords

Motif Discovery, Subsequences, Clustering, Pattern, Motif

1. INTRODUCTION

Motifs are short, conserved subsequences that are part of a family of sequences. The use of protein sequence patterns (or motifs) to determine the function of proteins is an essential tool for sequence analysis. The sequence of an unknown function might not be closely related to any protein of known structure to detect similarity by overall sequence alignment, but it can be found sometimes more accurately by the occurrence in its sequence of a particular cluster of residue types a.k.a pattern, motif, signature, or fingerprint. Some regions are conserved because of particular requirements on the structure of specific region of a protein which may be important, for example, for their binding properties or for their enzymatic activity.

Motifs can be discovered as subsequences that are common to the family of sequences from sequence patterns (subsequences). Since the dramatic increase of genetic data, clustering and motif finding techniques have become essential to the analysis of protein and nucleic acid sequences. As a result, a number of different clustering algorithms and motif discovery algorithms have been developed. Although these two techniques are being well studied, little work has been done to combine them and make an optimized motif discovery tool for general sequence analysis purposes. In this paper, we demonstrate how clustering of sequences can be used for motif prediction.

1.1 Background

Existing motif discovery algorithms can be classified into two groups, one group of algorithms, such as PRATT[4] and TEIRESIAS[9], that search for motifs by combinatorial enumeration of patterns and another group of algorithms, such as MEME[1] and GIBBS[6], that search for motifs that are statistically significant. Our interest in this paper is to further develop the second group of algorithms that search for statistically significant sequence patterns. Among them, Gibbs and MEME are the most widely used ones. Although these algorithms have been successful in predicting biologically meaningful sequence patterns, no algorithm works perfect as the motif discovery problem is very difficult to solve. The problem with MEME is that it takes a lot of time to find motifs because it uses multiple rounds of expectation maximization technique in search of motifs. The Gibbs algorithm has a random search behavior and each execution generates different motifs.

A more serious problem is that motif discovery algorithms do not perform well when the input set consists of non-homogeneous sequences. In particular, it is a challenge for all existing motif discovery algorithms to detect motifs when

BIOKDD '05, August 2005, Chicago, Illinois, USA

the input set contains sequences from multiple families since a motif occurs only in a subset of the input sequences and the number of motifs to be discovered is not known a priori. The problem can be stated as follows.

In a statistical sense, motif discovery is to look for signals compared to noise in the input sequences.

$$\log(\frac{M_{motif}}{M_{noise}})$$

where M_{motif} is a model for the motif and M_{noise} represents the model for background noise.

Different input sequences affect both M_{motif} and M_{noise}. M_{motif} requires selection of subsequences that will construct candidate motifs and M_{noise} is largely determined by the input sequences. More specifically, both models are character frequencies at specific positions as shown in Table 2 and these character frequencies are determined, with some prior knowledge, by counting the number of characters in a specific column. Thus they are directly influenced by selection of subsequences and the input sequence set.

1.2 Research Question & Motivation

The question that we are looking to answer here can be stated as follows -

Can there be an efficient tool that discovers motifs from a heterogeneous set of sequences when -

1. Width of each motif is not known
2. Number of motifs to be found is not known ?

We try to explore this problem by applying the techniques of clustering and applying it to motif discovery question. We have been successful in guiding motif discovery by clustering sequences[3] and by sampling subsequences[11]. In this paper, we combine these two approaches in a different context. Once subsequences are selected by a method proposed in [11](see also Section 2.1 in this paper for an improved subsequence selection method), it is assumed that there is a single motif. However, it is possible that there may be more than one motif in the set of subsequences. To deal with this problem, we used a sequence clustering algorithm inspired by our previous approach[3]. Instead of clustering the input sequences directly as in, a set of subsequences of equal length are generated and then clustered using our clustering algorithm as described in Sections 2.2, 2.3, and 2.4. This procedure does not require information about how many motifs need to be sought, which is a unique feature of our motif algorithm.

2. METHODOLOGY

Given a large data set of N protein sequences $S_1, S_2, \ldots, S_N$, the goal is to identify the conserved regions that represent this data set. Our algorithm is designed to proceed in the following manner -

1. Extract an initial set of patterns.
2. Select subsequences and align them.
3. Cluster those subsequences.
4. Extract conserved regions from shared ranges for each cluster.
5. Extend/Merge the conserved regions.
6. Rank those conserved regions based on information content.
7. Filter and select motifs.

2.1 Initial Pattern Discovery

From the set of input sequences, a set of initial patterns is collected. The set of initial patterns are exact patterns of a fixed length l that satisfy two conditions -

1. Patterns are statistically significant.
2. Patterns are present in certain number of input sequences.

In our experiments, l is fixed to 10 which is an empirically determined value - exact patterns longer than 10 do not occur frequently even in the conserved motif regions. The statistical significance of patterns is required since motifs will occur significantly more frequently than random patterns, which is the basis for most statistical motif discovery algorithm. The second condition is required since motifs are recurring patterns common in multiple sequences; patterns that occur multiple times in a single sequence but do not occur in any other sequences are not qualified.

In addition to the two conditions, the set of patterns should represent all input sequences while multiple different patterns can be sampled around motif regions. For this reason, we use a procedure that iteratively discards the top half of sequences where patterns are already sampled while looking for patterns that meet the two conditions as described in Section 2.1.2.

2.1.1 Two Conditions for Patterns.

The condition for statistical significance of a pattern in our previous paper was based on the first order Markov model. We improve this by using the second order Markov model. The challenge is that it is difficult to measure the second order dependency for a short pattern. For example, only the third character, or after, in a pattern of length 6 can have two proceeding characters. Given that the second order dependency cannot be measured for the first two characters, it is not very effective to use the second order Markov model for a short pattern. Thijs et al[10] used characters proceeding the motif, i.e., those outside the motif, to compute higher order Markov dependency and used the higher order model as background model to improve the performance of the Gibbs motif algorithm. Inspired by this work, we modified our statistical significance condition to use the second order Markov model as follows.

Let x be a sequence of amino acids, e.g. $x = x_1x_2\ldots x_l$. The probability of x for a given second order Markov model M is

$$P_M(x) = \Pi_{i=1}^{l} P(x_i|x_{i-2}x_{i-1})$$

where $P(x_1|x_{-1}x_0) = P(x_1)$ and $P(x_2|x_1x_0) = P(x_2|x_1)$ if x_0 and x_{-1} are not available. The probability of x for a given random model R is $P_R(x) = \Pi_{i=1}^{l} P(x_i)$. Then the log-odd score of the sequence x, denoted $E(x)$, is defined as

$$E(x) = \log(\frac{P_M(x)}{P_R(x)})$$

The log-odd score can be found for any pattern $x_ix_{i+1}x_{i+2}$ of x by using the initial overall probabilities and setting

Table 1: Algorithm for Initial Pattern Discovery

```
Input: $S$                    // a set of sequences
Output: $P$                   // a set of patterns
$P_q = \phi$                  // a set of qualified patterns
$S_r = \phi$                  // sequence set represented by $P_q = \phi$
while($|S_r| < n$) {
    $S_r = S_r \cup$ qualified_pat($l, K, T, S - S_r$)
}
return $P_q$

qualified_pat($l, K, E, S'$) {
    Find $P_q$ (meet thresholds $l, K, T$) in sequence set $S'$ ;
    Rank $S'$ according to the number of $P_q$ in each sequence ;
    $S''$ top half of $S'$ ;
    for each qualified pattern $P \epsilon S''$,
        $P_q = P_q \cup \{P\}$ ;
    return $S''$
}
```

$l = 10$. We consider all patterns of length 10 in the input sequences and consider them statistically significant if their log-odd score is greater than threshold T, i.e. $E(x) > T$.

The second condition that patterns should be present in a certain number of sequences can be simply enforced by a support ratio K, the occurrence of a pattern in at least $K\%$ of the sequences. This avoids the case that one pattern occurs many times in one sequence, but rarely appears in other sequences. So the support value is set to make sure the qualified patterns are common features for the family.

2.1.2 Algorithm for Initial Pattern Discovery

At each iteration step, we rank the sequences according to the number of qualified patterns they have, eliminate the top half of the sequences and leave the rest half for the next iteration step. Since patterns are sampled only from sequences not represented by the current pattern set, it is ensured that all sequences are represented by some patterns. The algorithm used for initial pattern discovery is shown in Table 1. In the table, l denotes the length of the subsequence, K represents the support ratio as explained above and T is the log-odds threshold above which a subsequence is considered significant.

2.2 Subsequence Selection and Pairwise Alignment

The set of patterns selected using the procedure in Section 2.1 form the initial set of subsequences. These subsequences are aligned using standard scoring matrix (e.g. BLOSUM62). The scoring or weight matrix is a standard method for representing the variation in a set of sequence patterns in a multiple sequence alignment, and as a tool for finding additional sequences with the same pattern in a database search. The odds score at every possible matching location along the subsequence may be used to find the probability of each sequence location. FASTA algorithm[8] is used to align those subsequences. The FASTA algorithm runs quite fast and is ideally suited for aligning large number of initial subsequences.

2.3 Clustering of Subsequences

Those aligned subsequences are then grouped into different clusters using our sequence clustering algorithm[5].

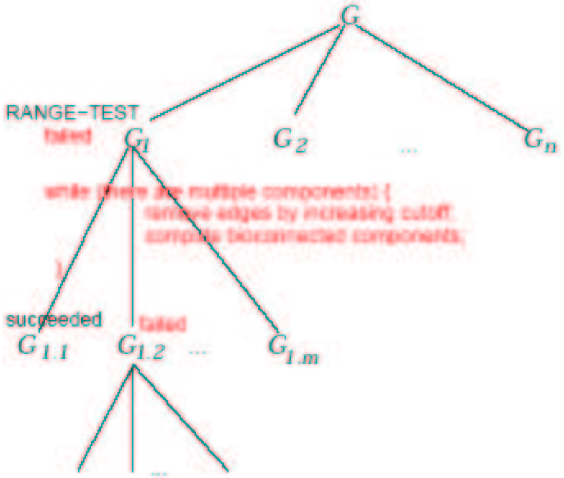

G
RANGE-TEST
failed
G_1
G_2
...
G_n
while (there are multiple components) {
remove edges by increasing cutoff
compute biconnected components;
}
succeeded
$G_{1.1}$
failed
$G_{1.2}$
...
$G_{1.m}$
...

Figure 1: The graph is iteratively refined until each biconnected component has a common shared region to all sequences in it.

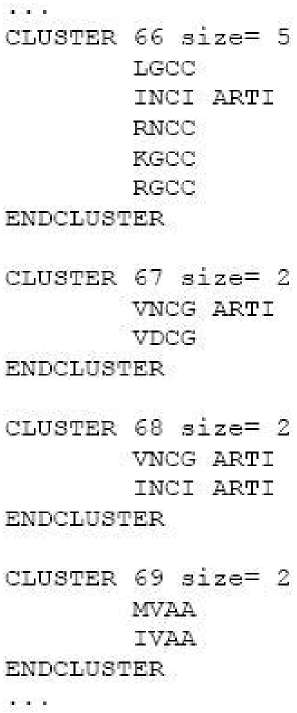
Given a set of sequences, our clustering algorithm builds a weighted graph based on similarities between those sequences. A node is created for each sequence s_i and an edge between two sequences, s_i and s_j, is created when the pairwise alignment score of s_i and s_j is more significant than a preset threshold. The alignment score is associated with the edge as weight so that clusters can be refined while increasing cutoff for edges. Our algorithm uses two graph properties for sequence clustering: *biconnected components* and *articulation points*. A biconnected component of a graph G is a maximal subgraph where there exist at least two edge disjoint paths for any pair of nodes, and an articulation point is a node that disconnects the graph if it is removed. A biconnected component corresponds to a family of sequences and an articulation point to a multi-domain protein. Each biconnected component is tested whether all sequences share a common shared region - this test is called RANGE-TEST. If there is no common shared region in a component, the cutoff score for the weight in the corresponding graph is increased until the graph is split into multiple biconnected components. The overall clustering procedure is depicted in Figure 1.

The set of input subsequences $\mathcal{S}$ is split into multiple clusters $C_1, C_2, \ldots C_n$ based on the sequence similarity among sequences in $\mathcal{S}$ using our clustering algorithm such that all sequences in each cluster have a common shared region of length L, by default $L = 6$ amino acids. In this way, subsequences are grouped together according to their sequence similarity defined by the scoring matrix, e.g., BLOSUM62.

A sample of the clusters generated by our sequence clustering algorithm is shown in Figure 2.

2.4 Extracting Conserved Regions

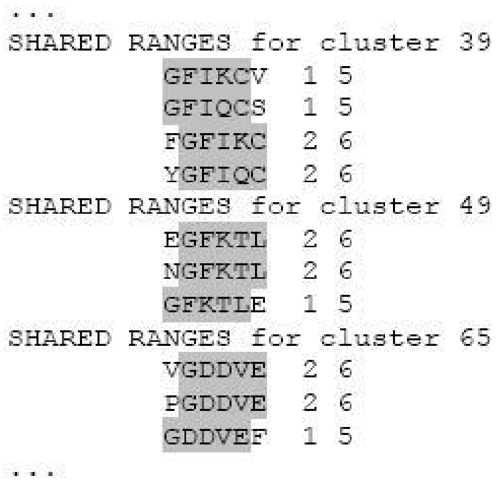
Our clustering algorithm produces clusters of these subsequences depending on their match score. However, we are not interested in all of these subsequences. The clustering algorithm also gives us a list of ranges which all the member subsequences of the cluster share. This eliminates all the clusters that are split and may not consist of conserved regions. Subsequence regions that are shared by all



```
...
CLUSTER 66 size= 5
        LGCC
        INCI ARTI
        RNCC
        KGCC
        RGCC
ENDCLUSTER

CLUSTER 67 size= 2
        VNCG ARTI
        VDCG
ENDCLUSTER

CLUSTER 68 size= 2
        VNCG ARTI
        INCI ARTI
ENDCLUSTER

CLUSTER 69 size= 2
        MVAA
        IVAA
ENDCLUSTER
...
```

Figure 2: Sample Clustering Output

the member subsequences of the cluster are extracted and these are the conserved regions that we are interested in. A sample output of shared ranges generated is shown in Figure 3. In the figure, shared regions of subsequences can be aligned without gaps to create a position weight matrix or a profile. Such shared regions can be determined by the start and end positions of the range produced by our clustering algorithm; the shared region starts at position 1 and ends at position 5 for the subsequence 'GFIKCV 1 5' of the cluster 39.



```
...
SHARED RANGES for cluster 39
          GFIKCV  1 5
          GFIQCS  1 5
          FGFIKC  2 6
          YGFIQC  2 6
SHARED RANGES for cluster 49
          EGFKTL  2 6
          NGFKTL  2 6
          GFKTLE  1 5
SHARED RANGES for cluster 65
          VGDDVE  2 6
          PGDDVE  2 6
          GDDVEF  1 5
...
```

Figure 3: Sample shared ranges. Grayed areas show the conserved regions.

An example of a motif model of length 10 is shown in Table 2. Given that sequence $x^i = x^i_1 x^i_2 \ldots x^i_W$ is the i^{th} motif with sequence length W, $P_{x_{ij}}$ represents the probability of j^{th} amino acid in sequence x_i occuring in the respective models.

Table 2: Example of a Motif Model of length 10. To save page space, only five character probabilities among 20 amino acids are shown for each colum.

Amino Acid	Position 1	Position 2	Position 3	...	Position 10
G	$P_{G,1}$	$P_{G,2}$	$P_{G,3}$	...	$P_{G,10}$
A	$P_{A,1}$	$P_{A,2}$	$P_{A,3}$	...	$P_{A,10}$
L	$P_{L,1}$	$P_{L,2}$	$P_{L,3}$	...	$P_{L,10}$
M	$P_{M,1}$	$P_{M,2}$	$P_{M,3}$	...	$P_{M,10}$
...	...	...	...	...	...
T	$P_{T,1}$	$P_{T,2}$	$P_{T,3}$	...	$P_{T,10}$

2.5 Merge/Extend the conserved regions

The length of the subsequences used for clustering puts an upper limit on the length of the motifs found above. It is possible that the regions around the found motifs may also be conserved. We take care of such situation by extending the found motifs in both the directions. The entropy of the original motif model is compared to that of the motif model after extension and if there is an increase in the information content, we extend the motif in that direction. This technique is similar to one used in CASTOR[7] pattern discovery program.

Let E_{motif} denote the log-odds score of the found motif, calculated from the motif model or profile M_{motif} of length W (see Section 2.1.1 for exact definition). Similarly, let E_{new} represent the log-odds score of the model $M_{motif+1}$ of length $W+1$ found after extending the motif model by one column. If $\frac{E_{new}}{E_{motif}} > \theta$, (where θ denotes the threshold ratio), then we accept that column as part of the consensus region and $M_{motif+1}$ forms the new motif model. This is repeated until there is no significant change in the information content.

It might be possible that two different motif models are overlapping or in close proximity to each other and we can merge such models. The strategy is to merge two motif models based on their correlation. The correlation between motif model A and B is given by -

$$\Gamma_{AB} = P_{AB}(d) - P_A P_B \tag{1}$$

where P_A, P_B are the individual model probabilities and $P_{AB}(d)$ is the probability of models A and B co-occuring at a distance d in the input set. So, if the correlation between two models is high, we merge them together to get a single motif model.

2.6 Ranking of Conserved Regions

Information content can be used to measure the degree of conservation at a site in a protein sequence alignment. The fewer the choice between occurrence of different residues at a site in the sequence, the more information it contains, and the more discriminatory it is for distinguishing real matches from random matches. The information content of the model can be expressed in terms of its entropy compared to that of the random model. The relative entropy H of a motif model M, as against the random model R, is

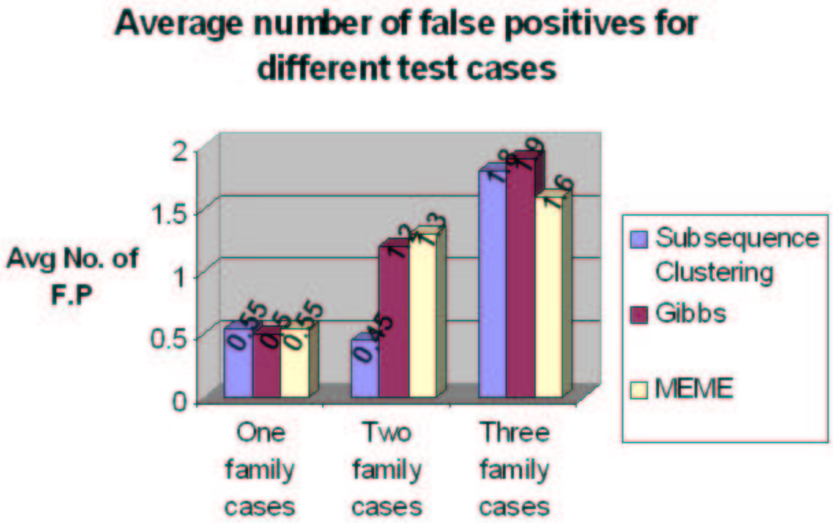


Figure 4: Average number of false-positives for various testcases

given by

$$H(M\|R) = \sum_{j=1}^{W} \sum_{x \epsilon AA} P^{M}_{x,j} \log(\frac{P^{M}_{x,j}}{P^{R}_{x,j}}) \qquad (2)$$

where $P^{M}_{x,j}$ and $P^{R}_{x,j}$ are the probability of the j^{th} nucleotide in x_i occurring in motif model M and random model R respectively, and AA is the alphabet of 20 amino acids. This favors longer sequence alignment because the information content will be high for longer sequences due to the first summation in the equation.

Besides, the support value of the models is also important because the motif should be able to represent a large number of input sequences. The support value (or *quorum*) represents the number of sequences covered by the motif.

The models corresponding to the conserved regions, extracted from the shared ranges, are ranked based on their support value (or *quorum*). More the support value, higher is the rank of the motif. The models covering the same number of sequences are further ranked based on their relative entropy.

2.7 Filter and select motifs

We further filter the predicted motifs based on their support weight. In a heterogenous dataset, one protein family might have more sequences than another. In order to reduce the bias towards larger families, we assign weights to each sequence and calculate the weight for each motif using

$$W = \sum_{i} w_i \qquad (3)$$

where W is the weight of the motif and w_i is the weight of sequence i, covered by the motif.

Initially all the sequences have similar weight(= 1). The support weight is calculated for each motif, starting from the top-ranked motif. If a sequence is covered by a motif, its weight is reduced by half. So the subsequent motifs covering the same sequences will have considerably lesser weight. We discard motifs whose support weight $W < 0.25N$ where N is the number of sequences in the input set. Filtering based on support weight helps to greatly reduce the number of false positives when there is a high representaion from one of the protein families. The motifs that remain are the actual motifs found by the algorithm.

3. EXPERIMENTS

In order to evaluate the correctness and efficiency of our proposed algorithm, the algorithm was applied on the collections of various PROSITE[2] protein families.

We consider three different scenarios for our experiments: Test family contains sequences from -

1. *single* protein family.

2. *two* different protein families.

3. *three* different protein families.

The protein families to be included in the test set is chosen randomly. For each of the testfamily, we try to find motifs using our subsequence clustering algorithm. To compare our algorithm against other established motif-discovery algorithms, we use Gibbs motif sampling algorithm[6] and MEME[1] on the same test set. Standard parameters are used for both gibbs and MEME; the number of motifs specified during the runs was 1,2 and 3 for one family, two family and three family scenarios respectively. Additional parameters for Gibbs and MEME -

1. Length of each motif = 15

Following parameter values were used in our algorithm -

1. Length of subsequences, l = 10

2. Threshold for finding subsequences, T = 0.01

3. Support ratio for finding subsequences, K = 0.05

4. Cutoff value for clustering algorithm, C = 100

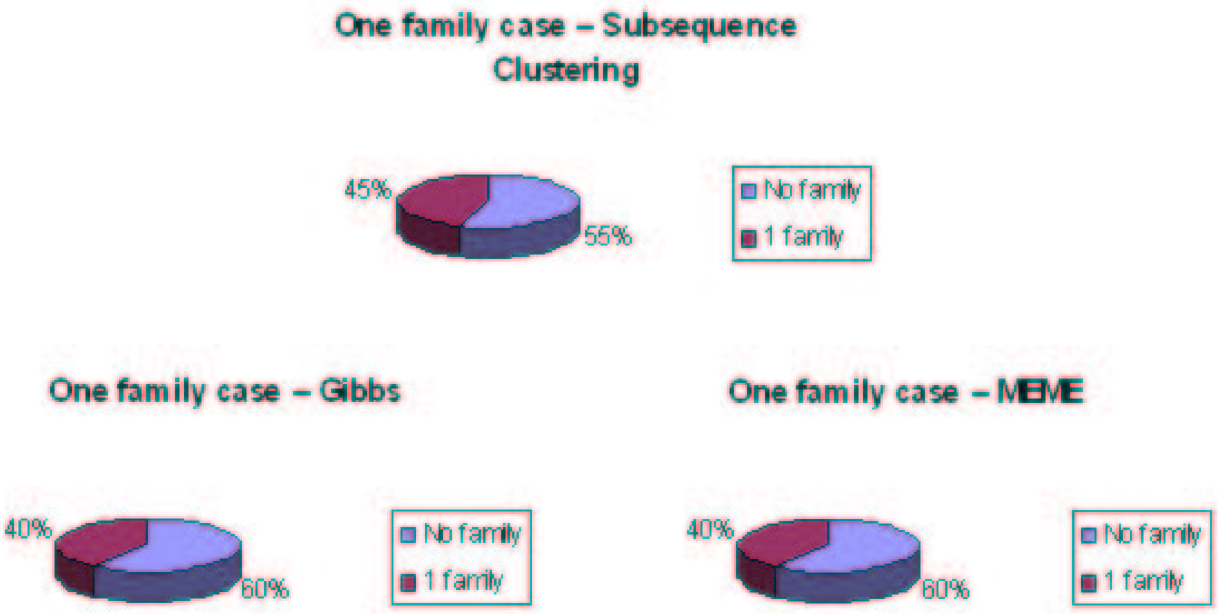


Figure 5: Number of families identified by (a) Subsequence Clustering approach, (b) Gibbs and (c) MEME for single-family testcases

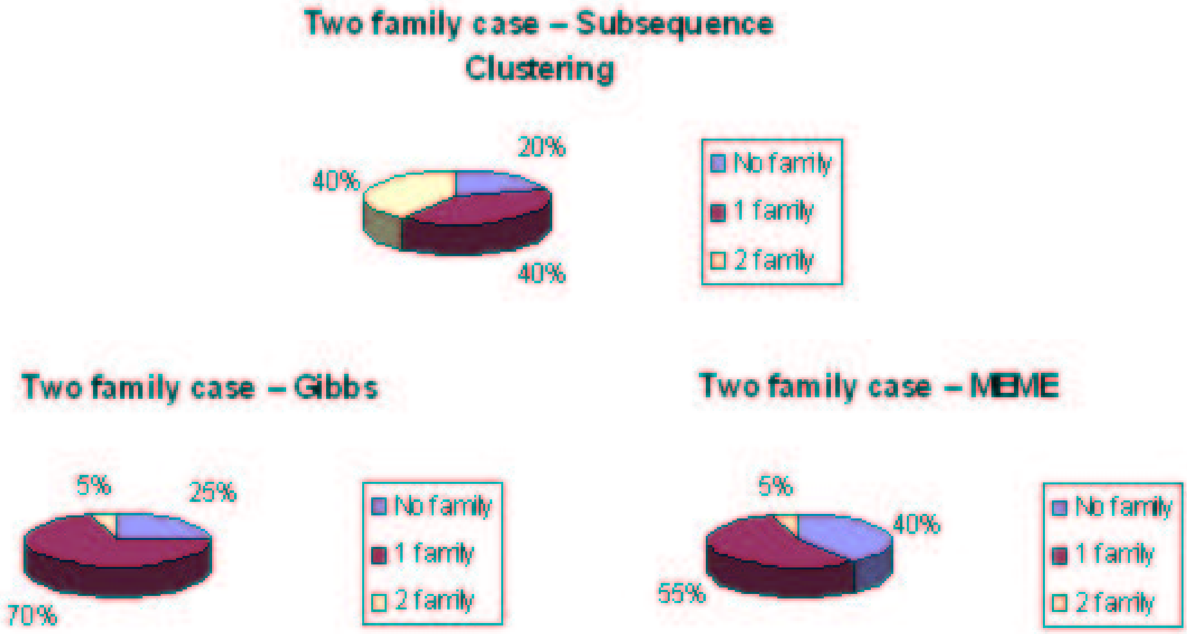


Figure 6: Number of families identified by (a) Subsequence Clustering approach, (b) Gibbs and (c) MEME for two-family testcases

5. Minimum overlap length for clustering algorithm, $L = 6$

We consider the motif found to be the actual motif if the predicted motif covers at least half of the PROSITE regular

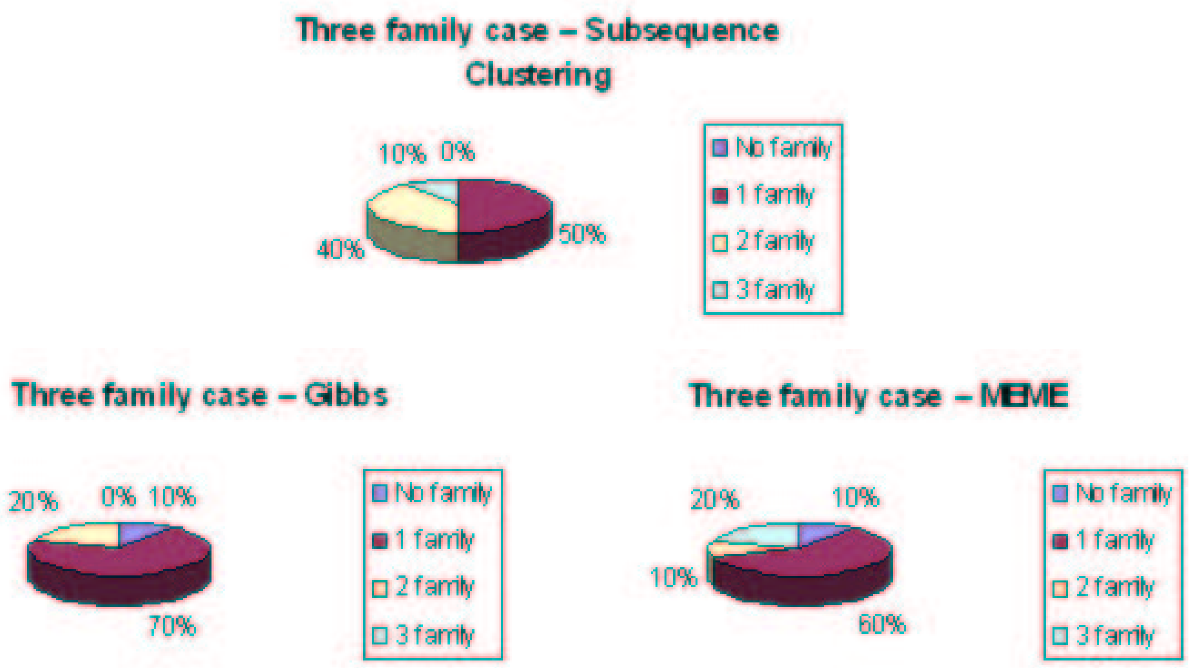


Figure 7: Number of families identified by (a) Subsequence Clustering approach, (b) Gibbs and (c) MEME for three-family testcases

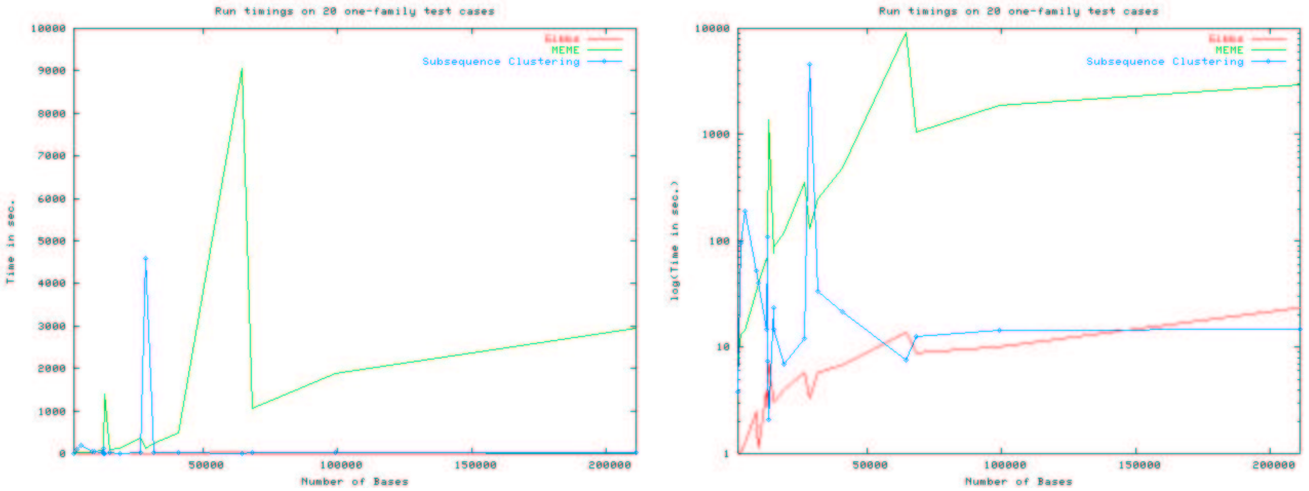


Figure 8: (a) Linear and (b) Semi-log plot of run timings for 20 one-family testcases based on the number of bases

expression. The results of our experiments for one family, two family and three family scenarios is shown in Figure 5, Figure 6 and Figure 7 respectively. We also compare the time taken by each algorithm to find motifs in the data set. The comparisons are shown in Figure 8 through Figure 10. The average number of false positives reported by the three algorithms is plotted in Figure 4.

4. DISCUSSION

As is evident from the graphs, the outcome of the three algorithms is comparable in case of single-family test cases. However, as the heterogeneity of the input set increases, results start showing a great deal of variation. The subsequence clustering approach is successful in discovering motifs representing the different families in more cases than Gibbs and MEME algorithm. In two family test scenario, our algorithm found motifs representing both families in 40% of the test cases, whereas Gibbs and MEME succeeded in only 5% of the test cases. The trend was similar in three family scenario, where gibbs couldn't find motif for all the

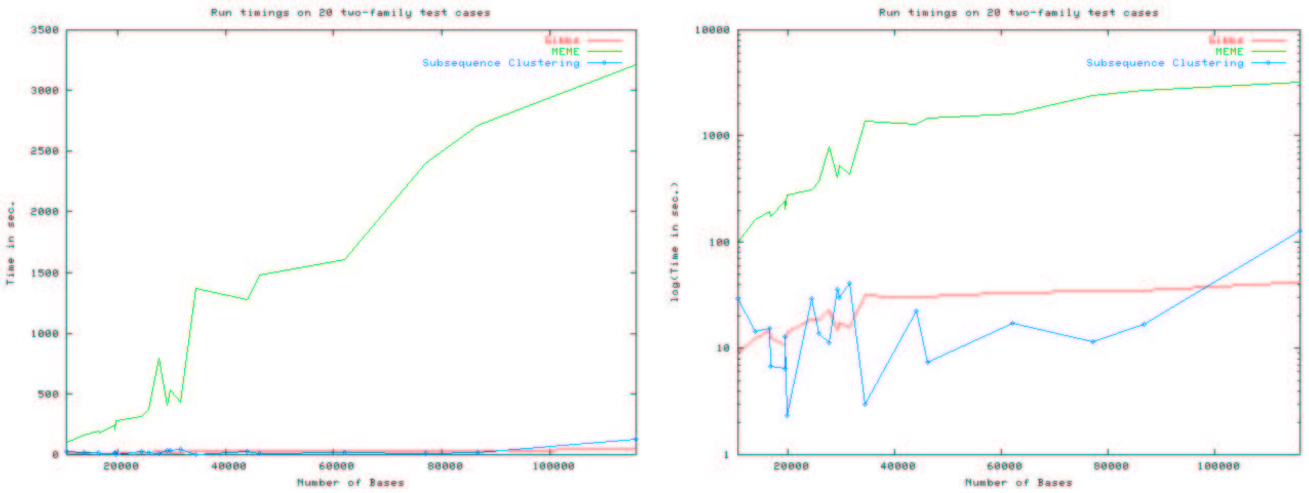


Figure 9: (a) Linear and (b) Semi-log plot of run timings for 20 two-family testcases based on the number of bases

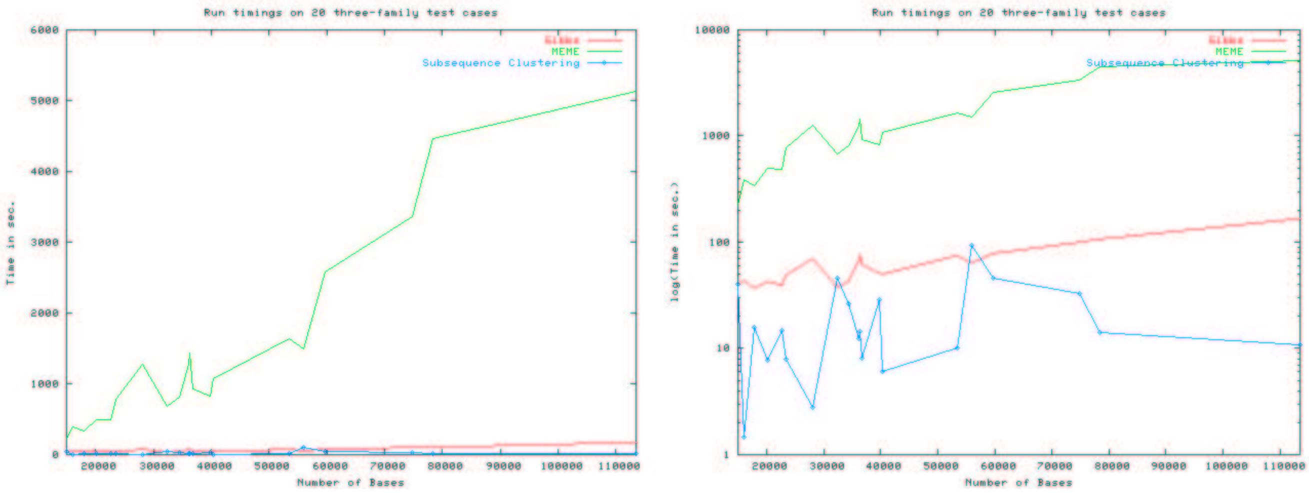


Figure 10: (a) Linear and (b) Semi-log plot of run timings for 20 three-family testcases based on the number of bases

three families in the input set in any of the test cases.

The average number of false positives produced by the three algorithms in different test cases is similar. However, we also emphasize that our method does not need to know the number of motifs expected. This is quite advantageous when there is no information available about the dataset.

As far as the runtime performance goes, our subsequence clustering algorithm works a lot faster than MEME. It is worth noting that subsequence clustering algorithm and gibbs were run on 4-processor Linux machine having 4GB of memory whereas MEME was run parallely on 2 nodes (8-processor) of IBM Scalable POWERparallel System (SP). The performance of our algorithm was comparable to gibbs in one family case and a little better in two and three family cases.

5. CONCLUSION

A motif discovery algorithm by clustering subsequences has been presented. The performance of our motif algorithm was measured using sequence families with patterns from PROSITE. The results of our algorithm were found to be comparable to those of gibbs and MEME motif algorithms and outperformed gibbs and MEME algorithms when input set contains non-homogeneous sequences, i.e., two family test cases. Our algorithm also outputs multiple conserved regions for the input sequence set without being instructed on how many motifs should be discovered. This is a very encouraging result and can prove to be very helpful in cases when the input sequence set consists of sequences from unknown protein families or from a collection of protein families. This is a significant advantage over existing motif discovery algorithms that essentially find only the specified number of motifs.

6. ACKNOWLEDGMENTS

We would like to thank Dr. Haixu Tang and Zhiping Wang for their support and helpful discussions on this project. This work is partially supported by NSF DBI-0237901, Outstanding Junior Faculty Award from Indiana Unversity and

INGEN (Indiana Genomics Initiatives).

7. REFERENCES

[1] T. Bailey and C. Elkan. Unsupervised learning of multiple motifs in biopolymers using em. *Machine Learning*, 21(1-2):51–80, October 1995.

[2] L. Falquet, M. Pagni, P. Bucher, N. Hulo, C. Sigrist, K. Hofmann, and A. Bairoch. The prosite database, its status in 2002. *Nucl. Acids Res.*, 30:235–238, 2002.

[3] I. Gunduz, S. Zhao, M. Dalkilic, and S. Kim. Motif discovery from large number of sequences: A case study with disease resistance genes in arabidopsos thaliana. *METMBS*, pages 29–34, 2003.

[4] I. Jonassen. Efficient discovery of conserved patterns using a pattern graph. *CABIOS*, 13:509–522, 1997.

[5] S. Kim. Graph theoretic sequence clustering algorithm and their applications to genome comparisons. *Computational Biology and Genome Informatics, World Scientific*, 2003.

[6] C. Lawrence, S. Altschul, M. Bogouski, J. Liu, A. Neuwald, and J. Wooten. Detecting subtle sequence signals: A gibbs sampling strategy for multiple alignment. *Science*, 262:208–214, 1993.

[7] A. H. Liu and C. Andrea. Castor: Clustering algorithm for sequence taxonomical organization and relationships. *Journal of Computational Biology*, 10(1):21–46, 2003.

[8] W. R. Pearson and D. J. Lipman. Improved tools for biological sequence comparison. *PNAS*, 85:2444–2448, 1998.

[9] I. Rigoutsos and A. Floratos. Combinatorial pattern discovery in biological sequences: The teiresias algorithm. *Bioinformatics*, 14:55–67, 1998.

[10] G. Thijs, M. Lescot, K. Marchal, S. Rombauts, B. D. Moor, P. Rouz, and Y. Moreau. A higher-order background model improves the detection of promoter regulatory elements by gibbs sampling. *Bioinformatics*, 17:1113–1122, 2001.

[11] Z. Wang, M. Dalkilic, and S. Kim. Guiding motif discovery by iterative pattern refinement. In *SAC '04: Proceedings of the 2004 ACM symposium on Applied computing*, pages 162–166, New York, NY, USA, 2004. ACM Press.

Graphical Models of Residue Coupling in Protein Families

John Thomas
Dept. of Computer Science
Dartmouth College
Hanover, NH 03755
jthomas@cs.dartmouth.edu

Naren Ramakrishnan
Dept. of Computer Science
Virginia Tech
Blacksburg, VA 24061
naren@cs.vt.edu

Chris Bailey-Kellogg
Dept. of Computer Science
Dartmouth College
Hanover, NH 03755
cbk@cs.dartmouth.edu

ABSTRACT

Identifying residue coupling relationships within a protein family can provide important insights into the family's evolutionary record, and has significant applications in analyzing and optimizing sequence-structure-function relationships. We present the first algorithm to infer an undirected graphical model representing residue coupling in protein families. Such a model, which we call a residue coupling network, serves as a compact description of the joint amino acid distribution, focused on the independences among residues. This stands in contrast to current methods, which manipulate dense representations of co-variation and are focused on assessing dependence, which can conflate direct and indirect relationships. Our probabilistic model provides a sound basis for predictive (will this newly designed protein be folded and functional?), diagnostic (why is this protein not stable or functional?), and abductive reasoning (what if I attempt to graft features of one protein family onto another?). Further, our algorithm can readily incorporate, as priors, hypotheses regarding possible underlying mechanistic/energetic explanations for coupling. The resulting approach constitutes a powerful and discriminatory mechanism to identify residue coupling from protein sequences and structures. Analysis results on the G-protein coupled receptor (GPCR) and PDZ domain families demonstrate the ability of our approach to effectively uncover and exploit models of residue coupling.

Keywords

Residue coupling networks, graphical models, evolutionary co-variation, sequence-structure-function relationships

1. INTRODUCTION

When studying a family of proteins that have evolved to perform a particular function, a major goal of contemporary biological research is to uncover constraints that appear to be acting on the family, with an eye toward understanding the molecular mechanisms imposing the constraints. For example, amino acid conservation has long been recognized as an important indicator of structural or functional significance [27]. In the 1990s, researchers began generalizing single-position conservation to correlated co-evolution of amino acid pairs, thus revealing cooperativity and coupling constraints (e.g., one early study focused on the HIV-1 envelope protein, with the aim of guiding peptide vaccine design [16]). Such works boosted the discovery of coupled residues, which could previously have been identified only by painstaking *in vitro* approaches such as thermodynamic double mutant analysis [11]. The next step was to summarize information about correlated positions into pathways [15], motifs [1, 20], and structural templates [20] in protein families. Today, projects undertake ambitious large-scale recombination [28] or site-directed and combinatorial mutagenesis studies [23] to identify entire building blocks of proteins important to preserve function.

Knowing which pairs (or sets) of residues are coupled in a protein family aids our understanding of many important processes, e.g., conformational change and protein folding [21, 24], signaling [26], protein-protein interaction, and even protein complex assembly [13]. Since the basis for coupling can be structural and/or functional, information about coupled residues can be used predictively for assessing protein structure [25], fold classification [9], or even to suggest novel sequences for protein engineering [22].

While there are many computational techniques for studying residue coupling [6], all methods begin by defining a metric to quantify the degree to which two residues co-vary. Global methods then determine pairs of coupled residues by observing correlated mutations in the protein family multiple sequence alignment (MSA) as a whole (e.g., [16]). The state-of-the-art in understanding residue coupling is, however, a local method—so-called 'perturbation-based' analysis [4] introduced by Lockless and Ranganathan [18]. The basic idea is to subset the MSA according to some condition (e.g., containing a moderately conserved residue type at a particular position) and observe the effect of the perturbation on residue distributions at other positions. If the subsetting operation significantly alters the proportions of amino acids at some other position, it is inferred to be coupled to the perturbed position, according to the evolutionary record. Even though this approach is purely sequence-based, it has been shown to uncover structural networks of residues underlying important allosteric communication pathways in proteins [26].

A key missing ingredient to date is a formal probabilistic model capturing the constraints inferred from residue coupling studies. Such a model would help assess the feasibility and significance of performing inference from coupling data, including determining whether coupling is a persistent feature of a protein family or merely a hallucination. The process of inferring such a model would help make explicit the essential constraints underlying the family (e.g., by identifying a small set of correlations that explain the data nearly as well as the complete set). A model would enable the careful combination of multiple information sources (e.g., by integrating priors from structural and functional studies with correlations derived from sequence analysis). Finally, the model would serve as a compact description of the joint amino acid distribution, and could be used for predictive (will this newly designed protein be folded and functional?), diagnostic (why is this protein not stable or functional?), and abductive reasoning (what if I attempt to graft features of one protein family onto another?).

This paper addresses these needs by formulating and elucidating the natural correspondence between residue coupling (qualifying interdependence among residues) and a probabilistic graphical model (summarizing interrelationships between random variables).

1. We present the *first* algorithm to infer an undirected graphical model, which we call a *residue coupling network*, representing coupling relationships in protein families. We bring in ideas from the extensive literature on probabilistic models [3] to derive networks that are meaningful as indicators of joint variation of sequence positions and that also explain structural features of protein families.

2. Unlike current correlated mutation algorithms that are focused on assessing dependence (which can conflate direct and indirect relationships) we focus on assessing *independence* (which enables modular reasoning about variation). We thus derive more compact descriptions of underlying networks highlighting the most important relationships.

3. We demonstrate how hypotheses regarding possible underlying mechanistic/energetic explanations for coupling can be used as priors for computational model discovery. For instance, if we have reason to believe that coupling in a given family would be only between nearby residues, a representative contact graph can be utilized as a valuable prior, benefiting algorithmic complexity and ensuring biological interpretability of the results.

2. BACKGROUND: CORRELATED MUTATIONS AND RESIDUE COUPLING

We begin by providing some background about correlated mutations and how they are used as indicators of residue coupling. Typically, we are given a multiple sequence alignment (MSA) whose rows are the members of the family and the columns are the aligned residue positions. Thus the MSA can be thought of as a matrix A where the value in row s and column j refers to the jth residue according to sequence s. We ignore columns with more than 50% gaps ('gapful' columns) and ignore in the calculations below the remaining entries that are gaps.

A coupling constraint quantifies the degree to which two positions in the family co-vary. Given positions i and k, information about amino acid occurrences contained in the ith and kth column vectors of the MSA can be summarized into 20-element vectors of frequencies, or probability distributions $P(i)$ and $P(k)$. Essentially, this allows us to think of residue positions as random variables over a discrete sample space of 20 possibilities (recall that we ignore gaps). Coupling can then be estimated by many information-theoretic and statistical metrics; one example is the (global) *mutual information* between $P(i)$ and $P(k)$, given by:

$$MI(i,k) \equiv \sum_{i=1}^{20}\sum_{k=1}^{20} P(i,k)\log\frac{P(i,k)}{P(i)P(k)}$$

Notice that the mutual information is actually the KL divergence [19] between the distributions $P(i,k)$ and $P(i)P(k)$; it quantifies the margin of error in assuming that the joint distribution $P(i,k)$ is decomposable. $MI(i,k)$ is zero when the underlying distributions are independent and non-zero otherwise. Another way to think of $MI(i,k)$ is as the difference

$$MI(i,k) \equiv H(i) - H(i|k)$$

where $H(i)$ is the entropy of the random variable i and $H(i|k)$ is the entropy of the probability distribution $P(i|k)$. If $MI(i,k) = 0$, then knowing the value of k does not reduce our uncertainty about i. A high score of $MI(i,k)$ is typically used as an indicator of coupling [16].

There are other ways to quantify coupling, e.g., using covariances and correlations; see [6]. In contrast to global methods for assessing coupling, perturbation based methods assess coupling between i and k by first selecting the rows of A that have position i fixed to some residue and observing the effect of this *in silico* perturbation on $P(k)$ (notice the asymmetry in this approach). Once again, we can assess the difference between $P(k)$ (before) and $P(k)$ (after) using a variety of metrics [4], including mutual information.

All metrics suffer from estimation problems under high or low degrees of conservation. For instance, if position i is always alanine and position k is always glutamine, then $MI(i,k)$ would be assigned zero even though we have not observed any variation in either! Similar problems arise with residues that have low frequencies of certain amino acids. It is hence well-recognized that 'correlated mutation algorithms must favor an intermediate level of conservation' [6].

A typical use of a coupling study is to visualize the inferred constraints in order to guide further experiments and gain insights into the sequence-structure-function relationship. For example, couplings have been organized into pathways of allosteric communication through the protein [15]. The discovery of such pathways has recently been reinvigorated with the work of [26] where the authors perform perturbation-based analysis at numerous positions and subsequently 'cluster' the pairs of coupled residues; this procedure has been shown to yield sparse, connected networks in many protein families. Researchers have also used coupling constraints as a basis to infer the contact map, since coupled residues are known to often be spatially proximal. This is still a popular way to validate correlated mutation algorithms (e.g., see [4]). Others compare the constraints to known energetic couplings inferred from double mutant experiments [7].

3. LEARNING GRAPHICAL MODELS OF RESIDUE COUPLING

If coupled residues indeed capture meaningful relationships, then they must afford a probabilistic interpretation. That is our working hypothesis for this paper and helps highlight where all previous work falls short. All previous approaches to inferring networks from data do so by direct incorporation of couplings as dependences and, as is well known, such an approach cannot distinguish direct from transitive dependences. It is also clear that (in)dependence of random variables is a very conditional phenomenon: two random variables may be correlated, become uncorrelated in the presence of new evidence, become correlated again when given further evidence, and so on. This means that we must pay careful attention to conditioning contexts, especially when we employ perturbation-based correlated mutation algorithms.

Our proposed approach is to directly learn a *residue coupling network*, an undirected graphical model $N(\mathcal{V}, \mathcal{E})$ that represents the residue coupling relationships. Such a model encodes probabilistic independence between its vertices according to an interpretation such as:

- *Pairwise:* For every pair (a, b) of non-adjacent nodes, a is conditionally independent of b, given every other node;
- *Local:* A node is conditionally independent of all other nodes, given its immediate neighbors; or
- *Global:* If a set of nodes c separates a from b, then a is conditionally independent of b given c.

In asserting independence between a given pair of random variables (nodes), notice that the *Global* interpretation uses a smaller conditioning context than the *Local*, whose conditioning context is even smaller than the *Pairwise* interpretation. For this reason, if a network satisfies the *Global* property, then it will also satisfy the *Local* property. Similarly, the *Local* property implies the *Pairwise* property. Symbolically, *Global* ⇒ *Local* ⇒ *Pairwise.*

Concomitant with the above independence interpretations, we can equally think of a network as representing a factorization of the joint pdf of the random variables in $\mathcal{V}$ (residues):

$$P(\{\mathcal{V}\}) = \frac{1}{Z} \prod_{c \,\in\, \text{cliques}(N)} \phi_c(v_c) \tag{1}$$

Here, the ϕ_c are potential functions so that

$$Z = \sum_{v} \prod_{c \,\in\, \text{cliques}(N)} \phi_c(v_c) \tag{2}$$

normalizes their product into a probability measure. In Eq. 1 and Eq. 2, v denotes instantiations of the joint sample space of $\{\mathcal{V}\}$ whereas v_c denotes instantiations over only those random variables participating in the clique (c). The structure of the potential functions satisfies:

$$\prod_{c \,\in\, \text{cliques}(N)} \phi_c(v_c) = \frac{\prod_c P(v_c)}{\prod_{a \,\in\, \text{cliqueadj}(N)} P(v_a)} \tag{3}$$

In other words, the likelihood is given by the product of marginals defined over the cliques of N divided by the product of marginals defined over the clique adjacencies of N (cliqueadj, which could be nodes, edges, or general subgraphs). In this view, each potential of Eq. 1 is either a conditional or a joint marginal distribution. For instance,

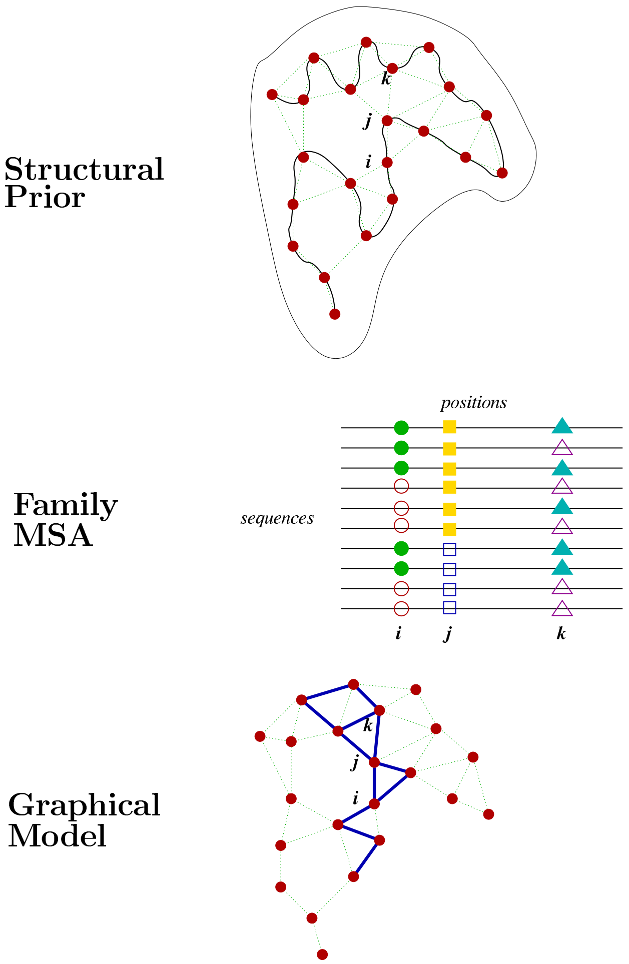


Figure 1: Residue coupling networks. (Top) A graph expressing a prior over possible coupling relationships. One source for a prior could be the contact graph representation of a protein's three-dimensional structure; here, mechanistic explanations for coupling posit either a direct interaction between contacting residues, or an indirect (transitive) propagation of an interaction through networks of contacting residues. (Middle) The multiple sequence alignment for members of a protein family provides evidence for dependence and independence. In the example, positions i and k are very correlated—when i is a 'filled in' residue, k tends to be as well; similarly when i is 'empty,' k tends to agree. However, knowing j makes the positions rather independent. In the most common case where j is filled in, we see the combinations of types at i and k are more evenly distributed. This suggests that i and k are conditionally independent, given j. (Of course, even in this example, noise obscures the degree of independence.) (Bottom) A graphical model (darkened edges) captures conditional independence. We construct such a model by selecting edges from the prior that best decouple other relationships. For example, we see that the conditional independence of i and k given j can be explained by a transitive propagation of interaction along model edges.

in an undirected network over three variables and two edges, with adjacencies (a, b) and (b, c), the product of the potentials is given by:

$$\phi_{a,b}\phi_{b,c} = \frac{P(a,b) \times P(b,c)}{P(b)}$$

We can view $\phi_{a,b}$ to be the conditional $(\frac{P(a,b)}{P(b)})$ and $\phi_{b,c}$ to be the marginal $(P(b, c))$, or vice versa.

Two well-known theorems in the probabilistic models literature [17] reconcile the independence and factorization viewpoints. First, if a distribution factorizes according to Eq. 1, then it satisfies the *Global* interpretation (and hence, the *Local* and *Pairwise* interpretations as well). Second, the Hammersley-Clifford theorem [3] states that if a joint pdf is positive everywhere (i.e., it has non-zero mass for all arguments), then it factorizes according to Eq. 1 iff it satisfies the *Pairwise* property (notice the bidirectionality of this theorem). Combining the above two theorems, we have: if a jpdf is positive everywhere, then the above three properties—*Pairwise*, *Local*, and *Global*—are equivalent. Any one of them holding true will imply the others.

In what follows, we adopt a statistical estimator of joint probability that assigns non-zero probability mass to every possible sequence. Thus, since the positivity assumption is satisfied, we can adopt any of the above three interpretations to infer independence between residue positions. In this case, the *Local* interpretation is easiest to operationalize. The *Pairwise* interpretation requires us to 'fix' (condition on) all but one residue and it is unlikely that this will retain a significant enough portion of the MSA to be confident about any probability assessments. The *Global* interpretation does not suffer from this drawback but makes the independence assessment more complicated by relying on a graph separation test.

If our MSA were sufficiently large and diverse enough to represent the joint probability of the family, then it is clear that the best unbiased estimator would be the maximum likelihood estimator (i.e., simply take the frequencies from the MSA). As the clique size grows, however, it is unlikely that the MSA is sufficiently representative of every possible clique value (i.e., set of residue types for the nodes). Therefore, we must consider the possibility that a clique value may not occur in the MSA but still be a member of the family. To this end, we adopt the following estimator for the probability of a clique value

$$P(c) = \frac{f(c) + \frac{\alpha N}{20^{|c|}}}{N(1+\alpha)} \tag{5}$$

Here $f(c)$ is the frequency of the clique value in the MSA, N is the total number of sequences in the MSA, $|c|$ is the size of the clique and α is a parameter that weights the importance of missing data. Notice that even when a particular clique value does not appear in the MSA, it still has a positive (but small) probability. This satisfies the desired positivity constraint. We are actively developing more sophisticated estimators, but results show that Eq. 5 is effective in practice. We employ a value of .1 for α but tests (data not shown) indicate that results are similar for reasonable values of α (between .01 and .25).

Uncovering graphical models from datasets is known to be an NP-hard problem in the general case and researchers typically restrict either the topology of the network (e.g., to trees [14]) or adopt heuristics to search the space of possibilities. In this paper, we assume the existence of a candidate set of edges (a graph prior; see below) and propose heuristics that sequentially infer conditional *independences* among this set (rather than dependences as followed in prior work). If we know that residues i and k become independent given j, i.e., the conditional mutual information

$$MI(i, k|j) = H(i|j) - H(i|k, j)$$

is zero, then it is easy to see that the removal of j and its incident edges must separate i and k in the unknown network N. This assessment is made in the context of a prior graph $G = (V, E)$, where we assume $\mathcal{V} = V$ and $\mathcal{E} \subset E$. This approach is akin to the 'sparse candidate' algorithm [8] for learning (directed) Bayesian networks.

```
function InferNetwork $(G = (V, E))$
$\mathcal{V} \leftarrow V$; $\mathcal{E} \leftarrow \emptyset$
$s \leftarrow \text{Score}(\mathcal{V}, \mathcal{E})$
$C \leftarrow \{(e, s - \text{Score}(\mathcal{V}, \mathcal{E} \cup \{e\}))|e \in E\}$
repeat
  $e \leftarrow \arg\max_{e \in E - \mathcal{E}} C(e)$
  $\mathcal{E} \leftarrow \mathcal{E} \cup \{e\}$
  for all $e' \in E - \mathcal{E}$ such that $e$ and $e'$ share a vertex
  do
    $C(e') \leftarrow C(e) - \text{Score}(\mathcal{V}, \mathcal{E} \cup \{e'\})$
  end for
until stopping criterion satisfied
```

Figure 2: Algorithm for inferring a residue coupling network.

Fig. 1 presents an example of such an inference. In attempting to de-couple position i from k, we need only consider neighbors of i (e.g., j) according to the graph prior. We consider here two priors: the complete graph or a contact graph. The complete graph is clearly an uninformative prior, assuming that all possible interactions are equally likely. The contact graph places edges between all pairs of residues that are "close-enough" (e.g., with some atoms within some distance threshold) in the three-dimensional structure of the protein. (Since structure is more conserved than sequence, we assume that all members of the family adopt essentially the same contact graph and select one from the PDB.) Physically speaking, this is a reasonable assumption in seeking to uncover direct energetic interactions and in distinguishing indirect ones propagated transitively (e.g., one residue 'pushes' another, which 'pushes' a third). We compare here results from these two priors, but note that other priors are possible, e.g., a graph accounting for functional information, coupling via an intermediate (ligand binding), or longer-range electrostatic coupling.

The score for a network, following the *Local* interpretation, is given by:

$$\text{Score}(N(\mathcal{V}, \mathcal{E})) = \sum_{n \in \mathcal{V}} \sum_{m \notin \text{neighbors}(n)} MI(n, m|\text{neighbors}(n))$$

In de-coupling a pair of positions i and k given neighbor j, rather than aiming for absolute independence $(MI(i, k|j) = 0)$, we assess by how much the conditional mutual information is decreased. We use the notion of network score to define an edge score as the difference in score between the network without the edge and the network with the edge.

Note that the score of an edge can be negative, if adding the edge produces more coupling in the network. Given the ability to evaluate the edges, we greedily grow a network by, at each step, selecting the edge that scores best with respect to the current network. Fig. 2 gives this algorithm. The algorithm can be configured to utilize various greedy stopping criteria—stop when the newly added edge's contribution is not significant enough, stop when a designated number of edges have been added, or stop when the likelihood of the model is within acceptable bounds.

The run-time of our algorithm depends on n, the number of residues in the protein of interest and d, the maximum degree of nodes in the prior. With an uninformative prior, d is n. For stronger priors (e.g., a contact graph), we can assume a bounded number of neighbors for any residue, so d is $O(1)$. The algorithm scores $O(dn)$ edges at each iteration. Naive execution of the algorithm requires that the score of the network be computed for each edge at each iteration. Scoring a network requires $O(n)$ MI computations for each residue and there are n residues, so naive execution requires $O(dn^3)$ MI computations at each iteration. Since conditioning contexts change dynamically during the operation of the algorithm, we cannot perform any *a priori* preprocessing to accumulate sufficient statistics (in contrast to global methods where mutual information between all pairs of residues can be computed in a single pass). However, the cost of making fresh assessments is curtailed since conditioning contexts are merely subsets of neighbors. Thus by caching values efficiently we can improve the runtime by a factor of $O(n^2)$ at each iteration. First, precompute the score of every edge in consideration, requiring $O(dn^3)$ MI computations. At each iteration, rather than recomputing scores, pick the edge in the cache that improves the score of the network the most. This requires $O(n)$ time, but does not require any MI computations. The key observation is that after an edge is added, the only edges whose scores change are those incident to the edge just added. Since there are at most $O(d)$ of those that need to be updated, we need only $O(dn)$ MI computations, for a speedup of $O(n^2)$. Additional constant factor speedups can be achieved by removing at each step edges that produce statistically unsound conditioning contexts.

4. EXPERIMENTS

We illustrate our algorithm for inferring residue coupling networks with two protein families: GPCRs (G-protein coupled receptors) and PDZ domains. GPCRs are membrane-bound proteins critical in intracellular communication and signaling, and a key target of molecular modeling in drug discovery. Since ligand binding at the extracellular face initiates propagation of structural changes through the transmembrane helices and ultimately to the cytoplasmic domains, GPCRs make an appropriate and compelling study for network identification [26]. PDZ domains are protein-protein interaction domains that occur in many proteins and are involved in a wide variety of biological processes [10]. One role of PDZ domains is assisting in the formation of protein complexes by binding to the C-termini of certain ligands [10]. Through these two studies we aim to explore many pertinent aspects of our approach, such as how to set priors, studying the progress of the algorithm as new edges are added, using the induced graphical model for classifying protein sequences, and biological interpretation of the results.

4.1 Results

4.1.1 GPCRs

In the GPCR study, we evaluate the use of protein contact graphs as priors and also explicitly relate the structure of our identified networks with those previously identified [26]. We first retrieved the multiple sequence alignment of 940 members of the class A GPCR family, each with 348 residues, as discussed in [26]. In order to explore contact graph priors, we constructed a contact graph from the three-dimensional structure of one prominent GPCR member, bovine rhodopsin (PDB id 1HZX), identifying 3161 pairs of residues with atoms within 7 Å. We verified that the residues previously identified as belonging to networks [26] form connected subgraphs of this contact graph.

For this study, in testing conditional mutual information, we only considered cases for which at least 15% of the original set of sequences remained after subsetting to a particular residue type. That is, we only allowed a residue to pick neighbors that, when restricted to their most common amino acid type, retain at least 15% of the original sequences. As discussed [18], such a bound is required in order to ensure sufficient fidelity to the original MSA and allow for evolutionary exploration. Our bound of 15% is roughly half that used in [26], since our algorithm subsets according to multiple residues, depending on the number of neighbors available, whereas the previous algorithm subsets according to only one residue. From extensive experiments with this parameter (data not shown), we found that while there is some variation in the edges with changes of this parameter, many ($> 70\%$) of the best edges are insensitive to the exact threshold.

In order to evaluate the implications of restricting dependences to structural neighbors, we compared the MI scores for edges in the protein contact map against those for all pairs of residues. This tested the hypothesis that the bulk of the correlation could be explained as correlation between structural (contact graph) neighbors. For every residue, we identified both the best decoupler *anywhere* in the protein, and the best decoupling contact graph neighbor. Fig. 3 shows the absolute differences between these values. Notice that in most cases, the best neighbor provides nearly as much decoupling as the best residue elsewhere in the graph. However, there are some nodes that incur a large penalty. In general, these nodes are highly conserved and therefore have small scores against all other nodes. However, since the total number of residues is large, the sum of all these small correlations becomes non-trivial. When a node is subsetted, making an originally highly conserved node become perfectly conserved, the score for that node drops to 0. In this case there is a large difference in improvement between selecting a distant node and a node from the original prior graph. It is important to keep these caveats in mind in the discussion that follows.

Our first model inference test was to start with the previously identified network of Suel *et al.* [26], use its induced subgraph of the contact graph as input to our algorithm, and see if we could recover the network. There are 144 edges to be considered. The algorithm constructed a model with 52 edges, after which point no other edge could be added without making the score worse, so the algorithm terminated.

Fig. 5 (left) illustrates the 52-edge network identified by our algorithm. Fig. 4 (red) shows the change in score as edges are added to the network. Notice the score decreases as edges are added and levels out toward the end (leading to termination when any remaining edge would increase the score).

To study the influence of the contact graph prior, we reran our algorithm using an uninformative prior so that all pairs of residues would be tested for inclusion. This time, the algorithm considered 1080 edges and picked 67 of them for inclusion before terminating with no edges available to improve the score. The resulting network has a better score than that of the network under the contact graph prior (Fig. 4 (blue)), but does not have as nice a visualization (Fig. 5 (right)).

Since the score differences between these two runs were substantial, we investigated the best possible score achievable for this protein family. Towards this end, we randomly shuffled the columns of the MSA, yielding a new MSA having the same level of conservation for each residue but with correlation lost due to the independent shuffling. We measured the correlation in 2500 of these MSAs (which consisted of just noise) by computing the score of the empty network (one with no edges) on the MSA. The resulting scores were normally distributed over a small range (63.5 to 65.1) with mean value 64.3. This means that for the GPCR family, if we accounted for all possible correlation we would expect a score of about 64.3. The algorithm run with the uninformative prior scores 73.6, well within the margin of error we would expect due to the greedy property of our algorithm or the nature of the conditioning contexts.

While our modeling formulation is different in nature from that of Suel *et al.* (independence vs. dependence, small number of parameters, etc.), our model that used the uninformative prior identifies many of the same biologically relevant features. For example, Suel *et al.* identify coupling between residues 296 and 265 that form "part of a linked network extending parallel to the plasma membrane from 296 to form the bottom of the ligand-binding pocket." Our algorithm likewise identifies an edge between residues 296 and 265. Several other identified interactions appear as *indirect* relationships in our model. For example, coupling between residue 296 and 293, identified as a "helical packing interaction" is identified by our model as being indirect. In this case, residue 117 actually makes residues 296 and 293 conditionally independent, lowering their mutual information scores from .3347 to .0259. This is true also of the coupling between residue 296 and residues 298 and 299. These couplings are part of "a sparse but contiguous network of inter-helical interactions linking the ligand-binding pocket with the cytoplasmic surface." Both 296/298 and 296/299 become conditionally independent in the presence of residue 117.

Although our algorithm does produce many of the relationships as identified by Suel *et al.*, there are several differences between the models. For instance, our network does not identify the coupling between residues 296 and 113 which "makes a salt-bridge interaction with the protonated form of the Schiff base," as either direct or indirect. Nor does our algorithm find the "inter-helical packing interaction" between residues 296 and 91. Conversely, our algorithm finds a strong direct coupling between residues 296 and 117 as well as between residues 90 and 91. Further investigation into these strong couplings may be of interest to biologists (e.g., by mutagenesis studies). This illustrates the ability of our approach to help formulate testable biological hypotheses.

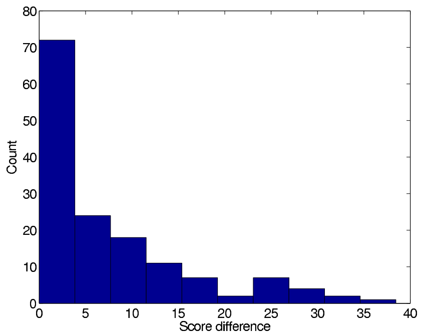


Figure 3: Penalty for decoupling using a contact graph neighbor rather than any residue (frequency distribution). Lower score differences indicate that neighbors perform as well as other residues.

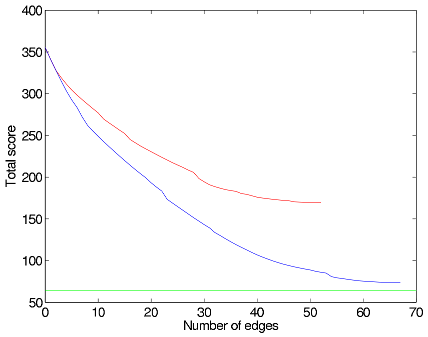


Figure 4: Improvement of MI score as edges are successively added for the contact graph prior (red) and uninformative prior (blue). The green line shows a lower bound for the score for the GPCR MSA.

4.1.2 PDZs

In the PDZ study, we demonstrate the utility in subsequent analyses of the graphical models learned by our algorithm. We study the ability of our inferred residue coupling networks to capture the 'essence' of a protein, namely in classifying PDZ domains. Traditionally, PDZ domains have been classified into two types according to which type of ligand they bind. The first class of PDZ domains binds to C termini with sequences S/T-X-Φ (Φ is a hydrophobic residue) while the second class targets sequences of the form Φ-X-Φ. Although the two classes in this protein family may be defined by simple sequence motifs, we show that coupling-based models provide more discriminatory power, and we use this opportunity to subject our approach to a rigorous evaluation in a maximum likelihood framework.

We obtained MSAs for the two classes of PDZ domains from PDZBase [2] by querying according to the ligand type and removing duplicate entries, thereby obtaining 95 class I and 12 class II sequences. We ran our algorithm on the

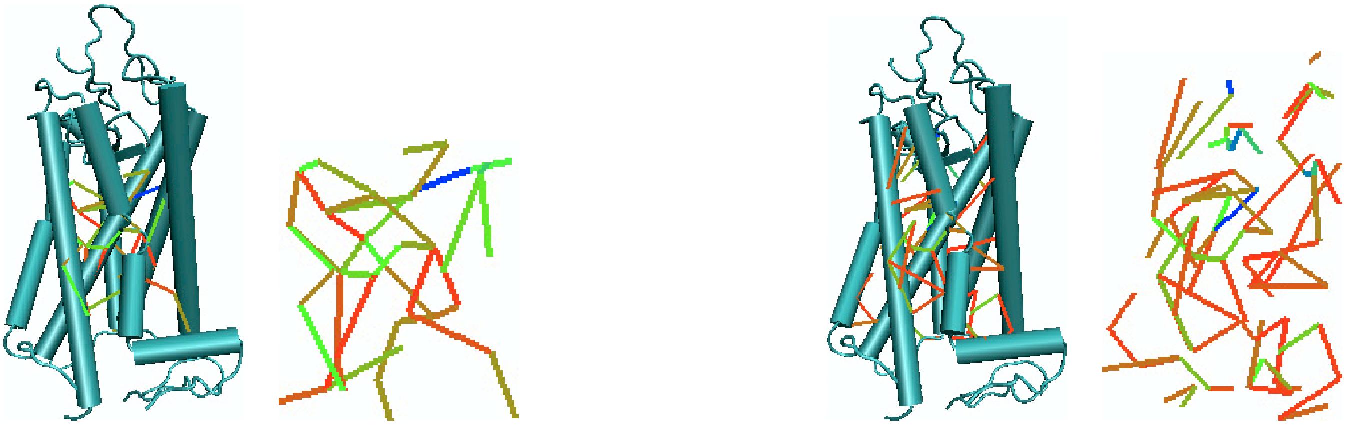


Figure 5: GPCR network identification: three-dimensional structure of bovine rhodopsin with overlaid network, and just the network for model inferred from (left) contact graph induced by the previously published network and (right) uninformative prior comprising all pairs of edges. Edges are colored by score, with red the strongest 'decouplers' and blue the weakest.

sequences in class I using an uninformative prior (no contact graph). After adding 85 of a possible 5671 edges to our model, the MI score converged (as was previously demonstrated with the GPCR family).

Using the estimator of Eq. 5, we compared the likelihoods from proteins in class I and II against different models, in a leave-one-out cross-validation test. Fig. 6 (top) shows the evolution of likelihood scores as edges are added to our model. On the far left of the plot is the likelihood based solely on conservation (i.e., with no edges in the network). As the network grows, so does its power to discriminate classes. Thus we conclude that conservation alone does not adequately represent the multiple sequence alignment. Once 40 edges are added to the network, the model has the power to discriminate perfectly between the two classes. We could continue to the limit by adding all edges to the network. In this case, we would derive a clique, with a joint distribution over all residues that would provide a reasonable score *only* for sequences in the original alignment. The convergence of the MI score prevents our algorithm from overfitting in this manner.

Fig. 6 (bottom) shows a receiver operating characteristic (ROC) curve that illustrates the classifying power of the conservation-based model and our inferred residue coupling network. The figure shows that classification of proteins can indeed be improved by moving beyond models that consider conservation alone to models that properly account for coupling relationships.

4.2 Comparison with Other Approaches

There are multiple dimensions along which our approach can be compared to others. The graphical models uncovered by our algorithm lie between a purely conservation-based representation of a protein family, and a dense representation of all co-variation within that family. As our results show quantitatively, we are able to account for the bulk of the co-variation with a significantly smaller number of parameters than is required by the complete graph assumed by other coupling studies. Thus our models should not overfit, but still account for significant coupling missed by pure conservation. Perhaps more importantly, while we employ the same co-variation analysis at the heart of our algorithm, none of the prior works results in a probabilistic model of any form, and hence none of them can systematically decompose observed co-variation into a core set of functional dependences, as is done here. This shortcoming holds even for the pioneering work on perturbation analysis [18, 26], since the 'networks' mined cannot be directly used as predictive models (e.g., from which new sequences belonging to the family can be drawn) or even as statistical indicators of variation (e.g., for assessing the likelihood of additional sequences). The approach presented here clearly overcomes these drawbacks by providing models that encode probabilistic assumptions of data and which can be genuinely falsified given appropriate data. We anticipate that this work will serve as a catalyst for more model-driven research into coupling networks.

5. DISCUSSION

This work marries research into residue co-variation with probabilistic graphical models, producing a systematic and sound algorithmic approach to inferring residue coupling networks underlying protein families. Our use of conditional mutual information as a criterion for growing a network means that our algorithm can also be viewed as a perturbation-based approach; however, in contrast to [26] who infer coupling between the perturbed position and another position, we infer independence between residues on either side of the perturbed position. The results indicate that independence of residues can be a good guiding principle for the discovery of evolutionarily conserved structure.

While there are other ways to infer networks from covariation data (e.g., gaussian graphical models [5]) they either require the specification of complete sets (e.g., all pairs) of dependency information or must necessarily make assumptions about the parametric form of interrelationships. In contrast, our approach employs the broader notion of independences to situate the network. In addition, it models *all* significant couplings and conditional independences, hence capturing the essence of what it means to belong to a given

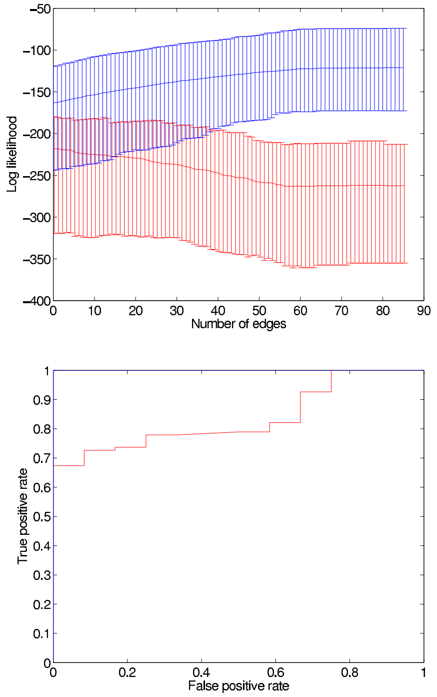


Figure 6: Evolution of likelihood as edges are added to the network. (Top) Sequences from class I (blue) and class II (red) against the class I model. Each plot shows the mean, maximum and minimum likelihood. The far left of the plot is the model based only on conservation. As the number of edges grows, more correlation is captured by the model. The far right is the model that contains all the correlations found by our algorithm. (Bottom) ROC curve showing the power of classifying by likelihood using only conservation (red) or the converged model produced by our algorithm (blue, following the box boundary).

family. This has tremendous applications in protein fold classification and protein design.

An important feature of our approach is the ability to make (selective) use of prior information towards a coupling study. Some priors (e.g., the contact graph) aid interpretability of the results but (as shown in our tests) might not yield as good as a model. There may be other potential explanations for observed couplings (e.g., electrostatics, ligand binding) that could be incorporated in the prior. Conversely, in the course of the algorithm, edges could be scored not only for reduction in MI but for consistency with a background theory.

The success of the approach is dependent on the quality of the provided MSA. We would like to scale up our algorithms to work with MSAs involving greater numbers of sequences, and thus more complete samplings of families. Inferring graphical models from such large datasets will benefit from research aimed at scaling up model inference (e.g., see [12]) and we propose to consider these for inferring coupled residues. We would also like to ensure fidelity of the alignment, particularly by using available structural information. Eventually, we hope to integrate alignment and model inference, perhaps employing shared hidden variables so that they iteratively improve each other.

Since motifs can be viewed as a limiting case (conservation only) of coupling relationships, we intend to build upon the work in that domain on representing general traits. For instance, we intend to relax our modeling of residues as distributions over amino acids, and instead consider distributions over *classes* of amino acids (e.g., polar, hydrophobic, small). Since there are multiple, overlapping, taxonomies of amino acids [27] we can even assume a hidden variable model (denoting an unknown relabeling of each residue) and attempt to infer the network as well as the relabeling function from a given MSA and contact map. An alternative is to employ a scoring matrix in evaluating extent of co-variation [24].

Finally, we intend to explore applications in protein design. Sampling from an inferred model is a natural way to generate new representatives of a family. Simultaneous construction of models for multiple families could help define their boundaries and thus even enable control over specificity in design.

Acknowledgements

CBK and JT are supported in part by a CAREER award from the National Science Foundation (IIS-0444544). NR is supported in part by NSF grants IBN-0219332 and EIA-0103660. This work was inspired by conversations with Drs. Rama Ranganathan (UT Southwestern) and Alan Friedman (Purdue). We thank Dr. Ryan Lilien (Dartmouth) for helpful comments on the manuscript.

6. REFERENCES

[1] W. Atchley, W. Terhalle, and A. Dress. Positional Dependence, Cliques, and Predictive Motifs in the bHLH Protein Domain. *Journal of Molecular Evolution*, Vol. 48:501–516, 1999.

[2] T. Beuming, L. Skrabanek, M. Niv, P. Mukherjee, and H. Weinstein. PDZBase: A Protein-Protein Interaction Database for PDZ-Domains. *Bioinformatics*, Vol. 21(6):827–828, 2005.

[3] W. Buntine. Operations for Learning with Graphical Models. *Journal of Artificial Intelligence Research*, Vol. 2:159–225, 1994.

[4] J. Dekker, A. Fodor, R. Aldrich, and G. Yellen. A Perturbation-Based Method for Calculating Explicit Likelihood of Evolutionary Co-Variance in Multiple Sequence Alignments. *Bioinformatics*, Vol. 20(10):1565–1572, 2004.

[5] M. Drton and M. Perlman. Model Selection for Gaussian Concentration Graphs. *Biometrika*, Vol. 91(3):591–602, 2004.

[6] A. Fodor and R. Aldrich. Influence of Conservation on Calculations of Amino Acid Covariance in Multiple Sequence Alignments. *Proteins: Structure, Function, and Bioinformatics*, Vol. 56:211–221, 2004.

[7] A. Fodor and R. Aldrich. On Evolutionary Conservation of Thermodynamic Coupling in Proteins. *Journal of Biological Chemistry*, Vol. 279(18):19046–19050, Apr 2004.

[8] N. Friedman, I. Nachman, and D. Peer. Learning Bayesian Network Structure from Massive Datasets:

The “Sparse Candidate” Algorithm. In *Proceedings of the Fifteenth International Conference on Uncertainty in Artificial Intelligence (UAI’99)*, pages 206–215, 1999.

[9] I. Grigoriev and S.-H. Kim. Detection of Protein Fold Similarity Based on Correlation of Amino Acid Properties. *Proceedings of the National Academy of Sciences, USA*, Vol. 96(25):14318–14323, Dec 1999.

[10] B. Harris and W. Lim. Mechanism and Role of PDZ Domains in Signaling Complex Assembly. *Journal of Cell Science*, Vol. 114:3219–3231, 2001.

[11] A. Horovitz. Double-Mutant Cycles: A Powerful Tool for Analyzing Protein Structure and Function. *Fold. Des.*, Vol. 1:R121–R126, 1996.

[12] G. Hulten and P. Domingos. Mining Complex Models from Arbitrarily Large Databases in Constant Time. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD’02)*, pages 525–531, 2002.

[13] A. Hung and M. Sheng. PDZ Domains: Structural Modules for Protein Complex Assembly. *Journal of Biological Chemistry*, Vol. 277(8):5699–5702, Feb 2002.

[14] D. Karger and N. Srebro. Learning Markov Networks: Maximum Bounded Tree-Width Graphs. In *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms (SODA’01)*, pages 392–401, 2001.

[15] I. Kass and A. Horovitz. Mapping Pathways of Allosteric Communication in GroEL by Analysis of Correlated Mutations. *Proteins: Structure, Function, and Genetics*, Vol. 48:611–617, 2002.

[16] B. Korber, R. Farber, D. Wolpert, and A. Lapedes. Covariation of Mutations in the V3 Loop of HIV Type 1 Envelope Protein: An Information Theoretic Analysis. *Proceedings of the National Academy of Sciences, USA*, Vol. 90:7176–7180, Aug 1993.

[17] S. Lauritzen. *Graphical Models*. Oxford University Press, 1996.

[18] S. Lockless and R. Ranganathan. Evolutionarily Conserved Pathways of Energetic Connectivity in Protein Families. *Science*, Vol. 286(5438):295–299, Oct 1999.

[19] D. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[20] M. Milik, S. Szalma, and K. Olszewski. Common Structural Cliques: A Tool for Protein Structure and Function Analysis. *Protein Engineering*, Vol. 16(8):542–552, 2003.

[21] O. Olmea, B. Rost, and A. Valencia. Effective Use of Sequence Correlation and Conservation in Fold Recognition. *Journal of Molecular Biology*, Vol. 295:1221–1239, 1999.

[22] W. Russ and R. Ranganathan. Knowledge-Based Potential Functions in Protein Design. *Current Opinion in Structural Biology*, Vol. 12:447–452, 2002.

[23] W. Sandberg and T. Terwilliger. Engineering Multiple Properties of a Protein by Combinatorial Mutagenesis. *Proceedings of the National Academy of Sciences, USA*, Vol. 90(18):8367–8371, Sep 1993.

[24] M. Saraf, G. Moore, and C. Maranas. Using Multiple Sequence Correlation Analysis to Characterize Functionally Important Protein Regions. *Protein Engineering*, Vol. 16(6):397–406, 2003.

[25] O. Schueler-Furman and D. Baker. Conserved Residue Clustering and Protein Structure Prediction. *Proteins: Structure, Function, and Genetics*, Vol. 52:225–235, 2003.

[26] G. Suel, S. Lockless, M. Wall, and R. Ranganathan. Evolutionary Conserved Networks of Residues Mediate Allosteric Communication in Proteins. *Nature Structural Biology*, Vol. 10:59–69, Jan 2003.

[27] W. Valdar. Scoring Residue Conservation. *Proteins: Structure, Function, and Genetics*, Vol. 48:227–241, 2002.

[28] C. Voigt, C. Martinez, Z.-G. Wang, S. Mayo, and F. Arnold. Protein Building Blocks Preserved by Recombination. *Nature Structural Biology*, Vol. 9(7):553–558, Jul 2002.

Predicting Cancer Susceptibility from Single Nucleotide Polymorphism Data: A Case Study in Multiple Myeloma

Michael Waddell
University of Wisconsin
Department of Computer Sciences Madison, Wisconsin, 53706
mwaddell@biostat.wisc.edu

David Page
University of Wisconsin
Department of Biostatistics and Medical Informatics
Department of Computer Sciences Madison, Wisconsin, 53706
page@biostat.wisc.edu

John Shaughnessy, Jr.
University of Arkansas for Medical Sciences Donna D. and Donald M. Lambert Laboratory of Myeloma Genetics Little Rock, Arkansas 72205
shaughnessyjohn@uams.edu

ABSTRACT

This paper asks whether susceptibility to early-onset (diagnosis before age 40) of a particularly deadly form of cancer, Multiple Myeloma, can be predicted from single-nucleotide polymorphism (SNP) profiles with an accuracy greater than chance. Specifically, given SNP profiles for 80 Multiple Myeloma patients – of which we believe 40 to have high susceptibility and 40 to have lower susceptibility – we train a support vector machine (SVM) to predict age at diagnosis. We chose SVMs for this task because they are well suited to deal with interactions among features and redundant features. The accuracy of the trained SVM estimated by leave-one-out cross-validation is 71%, significantly greater than random guessing. This result is particularly encouraging since only 3000 SNPs were used in profiling, whereas several million SNPs are known.

Categories and Subject Descriptors

J.3 [**Life and Medical Sciences**]: Miscellaneous

Keywords

supervised machine learning, support vector machines, single-nucleotide polymorphism, multiple myeloma

1. INTRODUCTION

A significant contribution to the genetic variation among individuals is the cumulative effect of a number of discrete, single-base changes in the human genome that are relatively easy to detect. These single positions of variation in DNA are called single nucleotide polymorphisms, or SNPs. While it is presently infeasible to obtain the sequence of all the DNA of a patient, it is feasible to quickly measure that patient's SNP pattern – the particular DNA bases present at a large number of these SNP positions [15].

Our case study employs support vector machines (SVMs) to analyze this new and promising form of genetic data. The authors present lessons for machine learning throughout the paper. Some biological terminology is necessarily used. Critical terms are defined for general machine learning (ML) readers; undefined terms are not critical to understand the ML lessons, but are used as needed to clarify issues for computational biology readers.

One promise of SNP data is that this data may make it possible to identify markers for genetic predisposition to disease. In addition to providing patients with information about their risk for disease, such markers may give researchers insight into the genes involved in a disease process and hence into proteins that may serve as targets for novel pharmaceutical therapies. In order to find such markers, the traditional approaches are to use linkage analysis and association studies [17].

Linkage analysis requires obtaining data on families with known pedigrees and disease histories. This requirement can make accurate linkage analysis difficult since many family members – including previous generations – are unavailable for genetic testing. Also, since the results of linkage analysis studies often come from a small number of families, they may not be generalizable to the rest of the population. Association studies do not require known family pedigrees. However, they do require a number of "candidate genes" that are suspected to be important in the disease process of interest. Thus, this method relies on the quality of the candidate genes, which are chosen based upon prior knowledge about the disease.

Both of these traditional approaches have been very successful when dealing with simple Mendelian or near-Mendelian disorders, but fail when attempting to identify disorders controlled by quantitative trait loci (QTL) [17]. QTL are genes, each of modest effect, whose combined effects cause a particular complex, continuous trait [5]. To deal with the complexities that QTL bring to this task, we will use an ML algorithm that is well suited to tasks involving interactions and redundant features.

BIOKDD '05, August 2005, Chicago, Illinois, USA
Copyright 2005 ACM 1-59593-213-5 ...$5.00.

First, we will divide the data points into two classes. Next, we will use an ML or statistical modeling algorithm to construct a classifier, or model, based upon all of the SNP data that were collected. The accuracy of the model at predicting the class (e.g., susceptible vs. not susceptible) will then be estimated using cross-validation. If the accuracy of the model is significantly better than chance, one may then study this model to gain insight into the disease. We have chosen not to employ candidate genes, like in an association study, because little is known about the genetics of Myeloma and its epidemiology. The hypothesis is that if there is an association between Myeloma and a particular gene, then a SNP in the haplotype block [4] containing that gene will be discovered in the present study. Given the general lack of knowledge about the etiology of this disease, we believe that using a candidate gene approach would put unreasonable bias on the analysis and, in the end, may fail and eventually cost more than doing a global search for associations.

This same general methodology has been employed in numerous cancer studies using microarray data [1, 6, 16, 18, 23]. A major advantage of using SNP data over microarray data to study genetic predisposition is that, unlike microarray data, a person's SNP pattern is unlikely to change over time. Loosely stated, the SNP pattern collected from a person with a disease is likely to be the same pattern that would have been collected from that person at birth or early in life. Thus, we can use SNP data from patients at any stage of their life and at any stage of their disease progression.

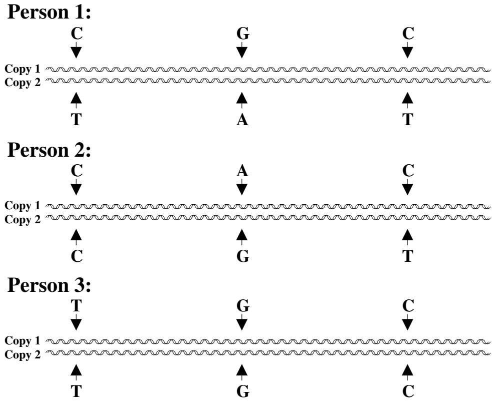
Single-nucleotide polymorphisms are extremely stable over evolutionary time [11]. Furthermore, relative to microsatellite polymorphisms, which are susceptible to mutations during the aging process [20], SNPs are much more stable and hence are unlikely to change over the lifetime of an individual [3]. The DNA used to perform our study is derived from peripheral blood mononuclear cells, which should be a mixture of cells whose germline DNA has no over-representation of any given clone containing any specific mutation. Thus, it is highly unlikely that the SNPs discovered in this study to be associated with the age of onset of Multiple Myeloma would be related to a SNP that tends to be mutated as a person ages. As a result of these arguments, SNP data has the potential to provide more insight into genetic predisposition to Multiple Myeloma, as well as many other diseases, than does microarray data.

A second major advantage of using SNP data is that the data can be collected from any tissue in the body. With microarray data, the mRNA samples for cancer patients are taken from tumor tissue (e.g., from the colon), and the mRNA samples for healthy donors are taken from healthy tissue of the same type (e.g., colon again). SNP data, on the other hand, is not taken directly from tumor samples, but from any tissue in the body. The benefit of this is that, in addition to being faster to obtain, SNP data is also easier to obtain since less invasive procedures can be used. On the other hand, when using SNP data, we do not expect to have predictors of as high accuracy as we get with microarray data. This is because microarray data is taken directly from the tumor tissue. Since gene expression is greatly altered in cancer, it is possible to obtain highly-accurate predictive models for cancer vs. normal. While such models may provide insight into the disease itself, they do not provide information on genetic predisposition. When working with SNP data, we expect to gain more information about a person's genetic predisposition to a disease than we would gain from microarray data; however, we do not expect to have predictors of as high accuracy as we get with microarray data.

Despite these advantages, SNP data does present three major challenges for our approach. The first challenge of SNP data is that there are now well over 1.8 million SNPs known [22], but measuring them all is typically cost-prohibitive. Hence, in contrast to microarray data where measurements are recorded for a substantial fraction of the known genes, SNP data contains measurements for only a small fraction of the known SNPs – typically a few thousand. Therefore, it is quite possible that, for a given classification task, the features that would allow for highly accurate prediction will be missing. Second, missing values are more common in SNP data than in microarray data. This must be taken into consideration when choosing a learning algorithm, since some methods are more capable of handling missing data than others. Third, and perhaps most interesting, SNP data is "unphased." Figure 1 illustrates this issue. The result of SNP data being unphased is that this additional, and potentially highly informative, phase information is not available for model building. Algorithms for haplotyping, or determining this phasing information, exist, but their solutions are not guaranteed to be correct. Also, these algorithms typically require additional data on related individuals and a large number of individuals relative to the number of SNPs [12]. Thus, one may approach this phasing problem either by estimating the phase information and accepting the consequences of incorrect estimates, or by working with the data in its unphased form. Because of the inaccuracies inherent in haplotyping and lack of additional data, we have elected to work with the data in its unphased form. We believe that this decision will not adversely affect our modeling algorithm since our research uses a relatively sparse coverage of the genome. Thus, adjacent SNPs are not linked strongly enough for phasing information to be informative. In future studies with a denser SNP coverage, this information would be potentially more useful.

Phasing, or haplotypes, are potentially informative because within a haplotype block there is very little, if any, meiotic recombination. Thus, the linkage of SNPs within a given haplotype block will remain unchanged over time. Once the haplotype map is established, it will be feasible to use a single SNP to define a haplotype block just as well as if one used all the SNPs within that block. It is estimated that there are approximately 600,000 haplotype blocks (there are currently some 300,000 defined) representing the millions of SNPs in the human genome [21]. These haplotype blocks may eventually be used to define the entire human genotype. When this occurs, haplotypes (defined by a single SNP) that are found to be linked to a disease could be searched for candidate genes and mutations within candidate genes. This will eliminate the guesswork that is inherent in the current candidate-based approaches which rely on an investigator's best guess or hunch.

This paper discusses the application of SVMs to SNP data in order to study genetic predisposition to Multiple Myeloma. Multiple Myeloma is a cancer of antibody secreting plasma cells that grow and expand in the bone marrow. Although Multiple Myeloma is hypoproliferative (the cancer cells replicate at a relatively low rate), the disease



(a) The true phased SNP patterns for persons 1, 2 and 3.

	SNP 1		SNP 2		SNP 3		Class
Person 1	C	T	A	G	C	T	Diseased
Person 2	C	C	A	G	C	T	Healthy
Person 3	T	T	G	G	C	C	Diseased

(b) The unphased SNP data for persons 1, 2 and 3.

Figure 1: In a SNP data file (b), each example, or data point, corresponds to a single person. The features, or variables, used to describe the person are the SNPs. A SNP position on one copy of a chromosome typically can take one of two values; for example, SNP 1 can be either C or T. But because every person has two copies of chromosomes 1 through 22, most SNP features can take one of three values. For example, the feature labeled SNP 1 can be either heterozygous CT as for Person 1, homozygous CC as for Person 2, or homozygous TT as for Person 3. If both SNP 2 and SNP 3 are on the same chromosome, then they can be arranged either as for Person 1 or for Person 2. Although these 2 arrangements are distinct, they lead to the same SNP pattern. The process of determining which of these two cases holds is called *phasing* or *haplotyping*. Data for which the haplotypes are not known is said to be *unphased*.

is incurable and usually progresses rapidly after diagnosis – with bone demineralization, renal failure, anemia, and secondary infections resulting from immunosuppression as common causes of mortality [19].

Multiple Myeloma occurs with relatively high frequency in adults over 70 (0.035% of the US population aged 70+) compared with younger adults (0.002% of the US population aged 30–54)[1]. We hypothesize that those who are diagnosed with Multiple Myeloma at a young age (under 40) have a genetic predisposition to the disease. If this is the case, then it may be possible to see differences in SNP patterns between Multiple Myeloma patients diagnosed before the age of 40 (predisposed) and those diagnosed after the age of 70 (not predisposed), and we can use these differences to gain insight into the disease. If this hypothesis is false, then it should not be possible to predict "predisposed" vs. "not predisposed" with accuracy significantly better than chance.

2. METHODOLOGY

Our data set[2] consists of unphased SNP data for 80 patients, based on 3000 SNPs, taking the form shown in Figure 1(b). The class values are "predisposed" and "not predisposed" as described at the end of Section 1. The 40 "predisposed" patients were diagnosed with Multiple Myeloma before age 40, while the 40 "not predisposed" patients were diagnosed after age 70. High molecular weight DNA was produced from peripheral blood lymphocytes from patients with Multiple Myeloma using conventional methods. DNA was subsequently sent to Orchid Biosciences^{TM}. SNP genotyping was performed using a proprietary SNP-IT^{TM}primer-extension technology. SNP-IT primer extension is a method of isolating the precise location of the site of a suspected SNP and utilizing the inherent accuracy of DNA polymerase to determine the allele type or the absence of that SNP. In order to conduct SNP-IT primer extension, a DNA primer (SNP-IT Primer) is hybridized to the sample DNA one base position short of the suspected SNP site. DNA polymerase is then added and it inserts the appropriate complementary terminating base at the suspected SNP location. Detection of the single base extension is accomplished by conventional methods. The result is a direct read-out method of detecting SNPs that creates a simple binary "bit" of genetic information. The SNPcode system couples SNP-IT genotyping technology with the Affymetrix GenFlex^{TM}platform to create a versatile, high-density SNP scoring system. In the assay, multiplex PCR is followed by solution phase SNP-IT primer extension. The SNP-IT products are then hybridized to the GenFlex chip – the sorting mechanism for the multiplexed reactions [14]. In the present study, 3000 SNPs were investigated on 80 patients. The SNPs were not selected based on prior knowledge of genetic disposition to Multiple Myeloma; rather, the SNPs were selected to give good overall coverage of the human genome. SNPs were chosen so that they would be evenly spaced at approximately every 1 megabase across the human genome. A denser coverage would be desirable but was cost-prohibitive.

We employed the approach of linear SVMs as our chosen modeling algorithm. We chose SVMs for this task because they are well suited to deal with interactions among features and redundant features. In particular, we used the algorithm SVM^{light} [9][3]. Because SVMs assume that all features are numerical, we needed to convert the discrete features from Figure 1(b) into continuous features. We will now present a brief review of SVM technology to help our readers understand the motivation behind our particular method of converting SNP features into numerical values.

In its simplest form, a support vector machine is an algorithm that attempts to find a linear separator between the data points of two classes, as Figure 2 illustrates. SVMs seek to maximize the margin, or the separation between the two classes, in order to improve the chance of making accu-

[1] Source: http://seer.cancer.gov

[2] The new SNP data set is available online from the authors at http://lambertlab.uams.edu/publicdata.htm.

[3] Publicly available at http://svmlight.joachims.org.

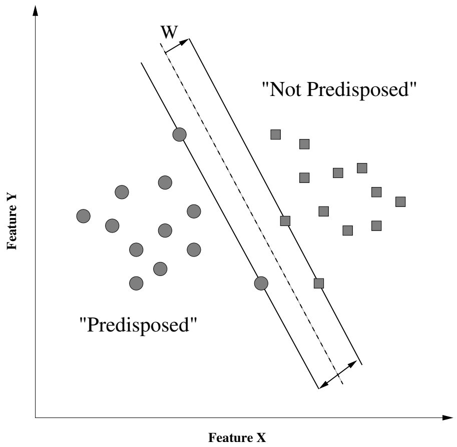


Figure 2: A support vector machine for differentiating between two classes by maximizing the margin, W. This is done in the N-dimensional space defined by N numerical-valued features. In this simple example, there are only two features, X and Y, so $N = 2$. Normally, however, N would be much greater. In a higher-dimensional space, the linear separator is a hyperplane rather than a line.

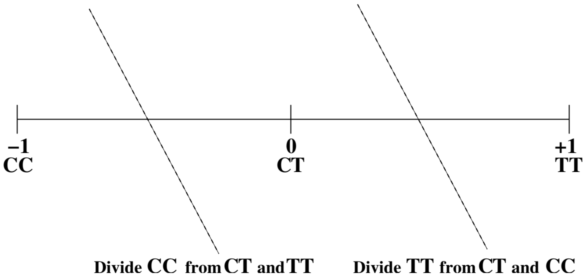


Figure 3: Divisions between feature values that are possible with the -1, 0, +1 encoding of SNP features. Notice that it is not possible to divide both CC (-1) and TT (+1) from CT (0) with a linear SVM.

rate predictions on future data. Maximizing the margin can be viewed as an optimization task solvable using linear or quadratic programming techniques. Of course, in practice there may be no good linear separator of the data. Support vector machines based on kernel functions can efficiently produce separators that are non-linear [2]. Nevertheless, the output of a linear SVM is easier to understand and glean insights from; effectively, features that get large coefficients in the function of the linear separator are more important than those that get small coefficients. In addition, linear SVMs have given better results than other kernel-based SVMs in several studies of microarray data, including our prior work with Multiple Myeloma. Therefore, for the present work we use linear SVMs. Experimenting with SNP data using other kernel functions is a direction for future work.

Each SNP feature in our data set takes one of three possible non-numerical values – either heterozygous or one of two homozygous settings (see Figure 1) – but SVMs require numerical features. Therefore we convert the three possible values for a SNP feature to the values -1, 0 and +1, where 0 represents heterozygous. We arbitrarily choose one homozygous case to set to -1 and the other to set to +1. As we see in Figure 3, when using this method with a linear SVM, it will be impossible to model the case where heterozygosity for a particular SNP is indicative of one class while homozygosity is indicative of the other, since it is not possible to separate 0 from both -1 and 1 with a single line. For example, it is not possible to say that either CC or TT is indicative of "predisposed" while CT is indicative of "not predisposed." Nevertheless, it is possible to distinguish having no copies of C from having at least one copy, or to distinguish having two copies of C from having zero or one copies (Figure 3).

Discriminating based upon the presence or absence of a single base appears to be more biologically relevant than discriminating solely based upon the presence or absence of homozygosity. In order for a heterozygous feature to not predispose cancer, whereas either of the two homozygous states do, the gene products of either allelic variant would be deleterious in sufficient quantities, but in the case of heterozygosity, neither would be present in sufficient quantities to cause a negative effect. In this case, regardless of the relative abundance of the two variants, a very large percentage of the population would be homozygous for one allelic variant or the other. Thus this feature would not be very informative and would not be incorporated into our model. In order for a heterozygous feature to predispose cancer, whereas either of the two homozygous states of that feature do not, the gene products of both alleles would need to be present to cause a negative effect. If both allelic variants were common in the general population, then heterozygosity of this feature would be relatively common and would thus not be very informative. If one allelic variant is relatively rare, then a homozygote in this feature will be very rare indeed. If such a rare person were to be found in our non-predisposed group, they would not likely affect our model significantly. Thus, it is very unlikely that the presence or absence of homozygosity would play a significant role in determining predisposition to a specific cancer. This supports our decision to use the absence or presence of a particular allele when building our model instead. This conclusion is further evidenced by the fact that most known mechanisms of inherited predisposition to cancers are dominant [10].

An alternative encoding that would permit all three possible distinctions between values would be to use two numerical features for each SNP. However, this leads to a doubling of the number of features, and the performance of ML algorithms tends to degrade as the number of features grows relative to the number of examples. Another option, using SVMs based on kernel functions, can efficiently produce separators that are non-linear [2]. Nevertheless, the output of a linear SVM is easier to understand and glean insights from; effectively, features that get large coefficients in the function of the linear separator are more important than those that get small coefficients. In addition, linear SVMs have given better results than other kernel-based SVMs in several studies of microarray data, including our prior work with Multiple Myeloma. Our preliminary studies using kernel functions to create a non-linear separator that *can* separate between the absence and presence of homozygosity have resulted in poorer performance than the simple linear separator. Further experimentation with SNP data using kernel functions is a direction for future work.

A major problem in ML applications is the "curse of dimensionality" – having many more features than examples. SVMs are more robust than some other ML algorithms when faced with high-dimensional data. Nevertheless, as with other ML algorithms, SVMs typically benefit from feature selection. Therefore, before training an SVM on our SNP data, we eliminate 90% of the features. Specifically, we select the top 10% (300) of the features according to information gain as described in the following paragraph. But before discussing the details of this approach, an important methodological point must be made. It is relatively common, though incorrect, to perform feature selection once by looking at the entire data set, and then to run cross-validation to estimate the accuracy of the learning algorithm. The resulting accuracy estimate is typically higher than will be achieved on new data, because the test data for each fold of cross-validation played a role in the initial feature selection process; hence information has "leaked" from the test cases into the training process. To avoid such an over-optimistic accuracy estimate, we repeated the following feature selection process on every fold of cross-validation, using only the training data for that fold. We chose to use cross-validation to assess the accuracy of our model since it is robust to high-dimensional data.

For each SNP feature we compute the information gain of the optimal split point, either between -1 and 0 or between 0 and 1. Information gain is defined as follows. The entropy of a data set is $-p\log_2 p - (1-p)\log_2(1-p)$ where p is the fraction of examples that belong to class "predisposed" (either class could have been used). A split takes one data set and divides it into two data sets: the set of examples for which the SNP feature has a value below the split-point and the set of data points for which the SNP feature has a value above the split-point. The information gain of the split is the entropy of the original data set minus the weighted sum of entropies of the two data sets resulting from the split, where these entropies are weighted by the fraction of data points in each set. The SNP features are then ranked by information gain, and the top-scoring 10% of the features are selected. A natural variant to the preceding procedure would involve making *both* splits, the split between -1 and 0 as well as the split between 0 and +1, dividing the original data set into three instead of two. The entropy and information gain equations extend naturally to this case as well. We chose to use binary splits to rank features because the linear SVM that will use these features will effectively make binary splits for each feature.

3. RESULTS AND DISCUSSION

We tested the approach described in the previous section using leave-one-out cross-validation. The confusion matrix is shown in Table 1. This yields an accuracy estimate of 71%, which is significantly better than random guessing. While this accuracy is not nearly as high as the accuracies we have grown accustomed to seeing for prediction of cancer vs. normal from microarray data, it is nevertheless exciting given that this prediction is based only on SNP data, which does not change once the disease occurs, and given that we had a relatively sparse covering of the genome with only 3000 SNPs.

Table 1: Confusion Matrix. This table shows how the class values predicted by the SVM on the test cases relate to the actual class values. This yields an accuracy estimate of 71%.

		Predicted	
		Not predisposed	Predisposed
Actual	Not predisposed	31	9
	Predisposed	14	26

To assess the significance of this result, we performed a permutation test. Permutation testing assesses the dependency of a classifier to the specific data set that is was designed for. This method is commonly used in situations where data is limited to give an estimate on the error of a classifier [8]. We performed the permutation test by randomly permuting the labels – "predisposed" and "not predisposed" – among the patients and running the entire cross-validated learning process on this new dataset. This entire procedure was repeated 10,000 times. The accuracy of these 10,000 classifiers very closely fits a normal distribution. The results of this test can be seen in Figure 4 and illustrate that our result of 71% is significant at the $p < 0.05$ level using a two-tailed test of significance. A standard binomial test was also performed and also established significance of the 71% result at the $p < 0.05$ level (two-tailed).

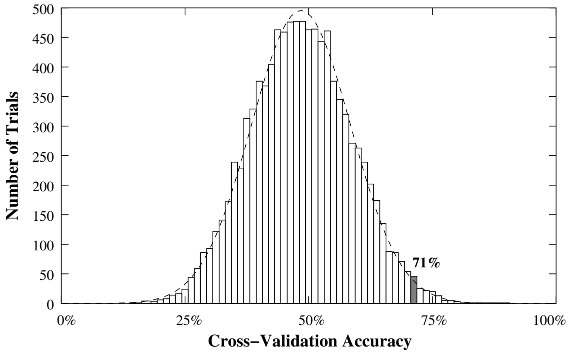


Figure 4: Results of a permutation test to estimate error of the classifier. We performed the permutation test by randomly permuting the labels – "predisposed" and "not predisposed" – among the patients and running the entire cross-validated learning process on this new dataset. This entire procedure was repeated 10,000 times. The accuracy of these 10,000 classifiers very closely fits a normal distribution. The 71% classifier is significant at the $p < 0.05$ level (two-tailed).

Although SNPs are highly unlikely to change within a single person as that person ages, it is true that certain SNPs will be underrepresented in certain age populations. For instance, a SNP that is associated with a gene responsible for causing a massive heart attack at age 50 will be present in a much higher proportion of 40-year-old patients than of 70-year-old patients. This emphasizes the need for the model that we build to be interpretable so that we can examine the SNPs that the model uses for prediction and determine their potential role in the disease mechanism.

In order to show that our learning algorithm is not basing its model on the age of the patients, we obtained SNP data on 28 unrelated persons without Myeloma from the SNP consortium[4]. 14 persons were older than 70 years-of-age and 14 were younger than 40 years-of-age at the time of SNP analysis. For each person, 2911 SNPs were chosen to provide broad genome coverage [13], just as the 3000 SNPs used with our "predisposed" and "not predisposed" patients were. Using the exact same procedure as we used for the "predisposed" and "not predisposed" data, we built a model using SVM^{light} after feature selecting the top 10% of features and using leave-one-out cross validation. The resulting accuracy was 46% and the confusion matrix can be seen in Table 2. Although the 2911 SNPs chosen were a different set of SNPs than the 3000 used with our patients, we believe that this result does provide evidence that the 71% accuracy we are obtaining with our model is unlikely to be from merely predicting age well. Our future work will include obtaining SNP data on persons such as these 28 using the same set of SNPs to further validate this conclusion.

Table 2: Confusion Matrix for Control Data. This table shows how the class values predicted by the SVM on the test cases relate to the actual class values. This yields an accuracy estimate of 46%.

		Predicted	
		Over 70	Under 40
Actual	Over 70	6	8
	Under 40	7	7

From the data in Table 1, we can compute the true positive and false positive rates for our model. The true positive rate is calculated as the fraction of the "predisposed" patients who are correctly classified as "predisposed." The false positive rate is calculated as the fraction of the "not predisposed" patients who are incorrectly classified as "predisposed." Using this method, we see that our model has a true positive rate of 65% and a false positive rate of 22.5%. However, because Myeloma is relatively rare in the general population, a false positive rate of 22.5% would result in a large number of patients being misdiagnosed as "predisposed." This is because our model was built with the naïve assumption that both types of misclassification errors (classifying "predisposed" as "not predisposed" and classifying "not predisposed" as "predisposed") are equally bad. In order to have the freedom to vary the relative misclassification costs of these two types of errors, we have plotted a Receiver Operator Characteristic (ROC) curve. An ROC curve is a standard way of assessing the accuracy of a model at varying degrees of conservativeness. As we see in Figure 5, if we choose a more conservative model that bounds our false positive rate to 5%, we are still able to achieve a true positive rate of 42.5%. This is very encouraging considering the limited data on which this model was based.

From these results we conclude that SNP data does indeed provide predictive ability for cancer susceptibility. That is the primary conclusion of this paper. The next question is whether the resulting SVM model can provide any insight into the disease. Ideally the SVM model would be based on only one or a few SNPs; that is to say, all but a few SNPs would have coefficients of zero in the equation for the separating hyperplane. Unfortunately, the model gives over 150 SNPs with non-zero coefficients. The maximum cross-validation accuracy that can be obtained for this data-set using a single SNP alone (using this SNP as a single voting attribute instead of using an SVM) is 61%, which is obtained using SNP 739514; a SNP on chromosome 4 at a location of 150,853,009 bp from the telomere of the *p* arm. If we instead use the top 3 SNPs (as determined by information gain) in unweighted majority-voting, we can achieve 72.5% accuracy (using SNPs 739514, 521522, 994532). Investigation of the full list of 150 SNPs is under way, but at this point we cannot claim that the model has provided useful insight into the disease. Although SVMs can accurately model the relative significance of features and their interactions, compared to some other algorithms such as decision trees and naïve Bayesian networks, their models are not easily interpretable.

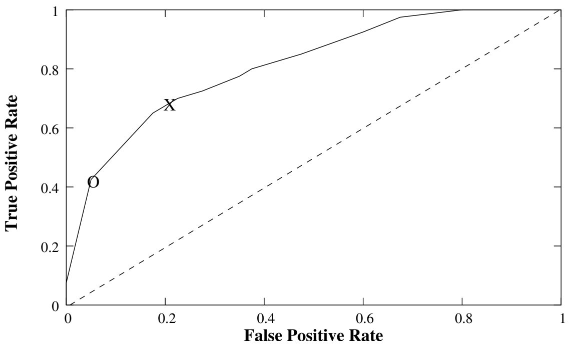


Figure 5: The ROC curve shows that linear SVMs (solid line) perform significantly better than random guessing (dotted line). It also shows the accuracy if we tune the SVM model to bound the false positive rate (since Myeloma is relatively rare in the general population). The point (5%, 42.5%) is noted with an *O*. The point without tuning (22.5%, 65%) is noted with an *X*. The true positive rate is calculated as the fraction of the "predisposed" patients who are correctly classified as "predisposed." The false positive rate is calculated as the fraction of the "not predisposed" patients who are incorrectly classified as "predisposed."

After finishing analysis of the linear SVM results, we reran our experiments using a few other standard ML algorithms. None of the algorithms that we tried – polynomial SVMs, decision trees (with and without boosting) and naïve Bayesian networks – performed significantly better than chance. Thus, we see that our choice of linear SVMs was a good one for this dataset and that the choice of algorithm can be very important when modeling biological datasets.

The only difference between linear and polynomial SVMs in this model is that polynomial SVMs are able to separate between the absence and presence of homozygosity (see

[4] http://snp.cshl.org

Figure 3) which, as we discussed in Section 2, is not biologically relevant. Thus, it is likely that polynomial SVMs were led astray by irrelevant correlations whereas linear SVMs were not able to be similarly led astray. Like polynomial SVMs, naïve Bayesian networks and decision trees are not well suited to this dataset. Because it appears likely that susceptibility to Myeloma is controlled by QTL and is not a simple Mendelian or near-Mendelian disorder, the feature independence assumption of naïve Bayes is strongly violated in our dataset. Decision trees are not robust with high-dimensional data and may have been led astray like polynomial SVMs since they too can separate absence and presence of homozygosity.

4. ONGOING AND FUTURE RESEARCH

Ongoing and future work is focused in three directions. First, we are cross-tabulating the SNP results with gene expression microarray results for Multiple Myeloma [7]. We are interested in whether any SNPs appear in or near genes that are differentially expressed in Myeloma vs. normal mRNA samples. We have found 11 SNPs that appear within 1Mbp of one of the top 1% informative (by information gain) genes for predicting Myeloma vs. normal from mRNA. We are also interested in whether any SNPs appear in or near genes that are differentially expressed in Myeloma vs. MGUS (a benign form of Myeloma) mRNA samples. We have found 7 SNPs that appear within 1Mbp of one of the top 1% informative (by information gain) genes for predicting Myeloma vs. MGUS from mRNA. We use a tolerance of ±1Mbp for two reasons. First, we see this breadth of deviation in SNP locations when using different information sources, e.g. NCBI and GeneCards. Second, research into haplotype blocks has revealed that large regions of DNA see very little recombination and tend to remain conserved, while recombination is largely isolated to certain "hot spots." Hence a SNP allele could be informative of a gene allele even if the SNP does not occur within the gene but only near it.

The second direction for ongoing and future work is to further tune the linear SVM algorithm as well as experimenting with other types of SVMs, such as Gaussian kernel SVMs (also available with SVM^{light}, for example), and with other types of modeling algorithms from ML and statistics. The goal of this work is to find a model for predicting predisposition for Myeloma that uses a smaller set of features for classification. This will allow us to gain a better insight into those regions that are important for conferring susceptibility.

Our final direction for future work is to repeat these experiments on a larger pool of participants, and using a denser coverage of SNPs, in order to further validate all of the findings of this study. We plan to do this in the next year or two when a sufficient number of the "predisposed" population (relatively rare) are referred to our center. In addition, we will look at the allele frequencies of the highly predictive SNPs in another similarly aged matched cohort.

5. ACKNOWLEDGMENTS

MW was supported in part by NLM grant 5T15LM007359. DP was supported in part by NSF grant 9987841 and by grants from the University of Wisconsin Medical School, Graduate School, and Comprehensive Cancer Center. JS and BB were supported in part by National Cancer Institute, Bethesda, MD grant CA55819. JS was also supported by a grant from the Fund to Cure Myeloma. We are grateful to Mark Craven, Christina Kendziorski and Jude Shavlik for helpful discussions of methodology and significance tests.

6. ADDITIONAL AUTHORS

Additional authors: Fenghuang Zhan (University of Arkansas for Medical Sciences, email: zhanfenghuang@uams.edu) and Bart Barlogie (University of Arkansas for Medical Sciences, email: barlogiebart@uams.edu).

7. REFERENCES

[1] M. P. S. Brown, W. N. Grundy, D. Lin, et al. Knowledge-based analysis of microarray gene expression data by using support vector machines. *P Natl Acad Sci*, 97(1):262–267, Jan 2000.

[2] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Min Knowl Disc*, 2:121–167, 1998.

[3] D. Burgner, K. Rockett, H. Ackerman, et al. Haplotypic relationship between SNP and microsatellite markers at the NOS2A locus in two populations. *Genes Immun*, 4(7):506–514, Oct 2003.

[4] S. B. Gabriel, S. F. Schaffner, H. Nguyen, et al. The structure of haplotype blocks in the human genome. *Science*, 296(5576):2225–2229, Jun 2002.

[5] A. M. Glazier, J. H. Nadeau, and T. J. Aitman. Finding genes that underlie complex traits. *Science*, 298(5602):2345–2349, Dec 2002.

[6] T. R. Golub, D. K. Slonim, P. Tamayo, et al. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, Oct 1999.

[7] J. Hardin, M. Waddell, C. D. Page, et al. Evaluation of multiple models to distinguish closely related forms of disease using DNA microarray data. *Stat Appl Genet Mol*, 3(1), June 2004.

[8] T. Hsing, S. Attoor, and E. Dougherty. Relation between permutation-test p values and classifier error estimates. *Mach Learn*, 52:11–30, 2003.

[9] T. Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1999.

[10] A. G. Knudson, Jr. Genetics of human cancer. *Annual Review of Genetics*, 20:231–251, 1986.

[11] R. Lewis. SNPs as windows on evolution. *The Scientist*, 16(1), Jan 2002.

[12] J. Li and T. Jiang. Efficient inference of haplotypes from genotypes on a pedigree. *J Bioinformat Comput Biol*, 1(1):41–69, Apr 2003.

[13] T. C. Matise, R. Sachidanandam, A. G. Clark, et al. A 3.9-centimorgan-resolution human single-nucleotide polymorphism linkage map and screening set. *Am J Hum Genet*, 73(2):271–284, Aug 2003.

[14] T. T. Nikiforov, R. B. Rendle, P. Goelet, et al. Genetic bit analysis: a solid phase method for typing single nucleotide polymorphisms. *Nucleic Acids Res*, 22(20):4167–4175, Oct 1994.

[15] M. Phillips and M. Boyce-Jacino. A primer on SNPs - part 1. *Innov Pharm Tech*, 1:54–58, Jan 2001.

[16] M. Ringnér, C. Peterson, and J. Khan. Analyzing array data using supervised methods. *Pharmacogenomics*, 3(3):403–415, May 2002.

[17] N. J. Risch. Searching for genetic determinants in the new millennium. *Nature*, 405(6788):847–856, Jun 2000.

[18] A. Rosenwald, G. Wright, W. C. Chan, et al. The use of molecular profiling to predict survival after chemotherapy for diffuse large-b-cell lymphoma. *New Engl J Med*, 346(25):1937–1947, Jun 2002.

[19] M. V. Seiden and K. C. Anderson. Multiple myeloma. *Curr Opin Oncol*, 6(1):41–49, Jan 1994.

[20] V. V. Symonds and A. M. Lloyd. An analysis of microsatellite loci in arabidopsis thaliana: Mutational dynamics and application. *Genetics*, 165:1475–1488, Nov 2003.

[21] The International HapMap Consortium. The international hapmap project. *Nature*, 426(6968):789–796, Dec 2003.

[22] G. A. Thorisson and L. D. Stein. The SNP consortium website: past, present and future. *Nucleic Acids Res*, 31(1):124–127, Jan 2003.

[23] L. J. Van't Veer, H. Dai, M. J. Van de Vijver, et al. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415(6871):530–536, Jan 2002.

Accelerating DNA Sequencing by Hybridization with Noise [*]

Chen Chen
Department of Computer Science
University of Illinois at Urbana Champaign, Urbana, IL 61801, USA
cchen37@uiuc.edu

Dong Xin
Department of Computer Science
University of Illinois at Urbana Champaign, Urbana, IL 61801, USA
dongxin@uiuc.edu

Jiawei Han
Department of Computer Science
University of Illinois at Urbana Champaign, Urbana, IL 61801, USA
hanj@cs.uiuc.edu

ABSTRACT

As a potential alternative to current wet-lab technologies, DNA sequencing-by-hybridization (SBH) has received much attention from different research communities. In order to deal with real applications, experiment environments should not be considered as error-free. Previously, under the assumption of random independent hybridization errors, Leong et al. [9] presented an algorithm for sequence reconstruction which exhibits graceful degradation of output accuracy as the error rate increases. However, as the authors also admitted, a notable downside of their method is its too high computational cost. In this paper, we show that the poor efficiency of [9] is due to its mixing-up of situations with widely different characteristics and treating everything in the safest but also slowest way. Our new algorithm addresses this problem and pushes analysis down to a finer level where a more effective solution is proposed. As demonstrated by experimentations on real human genome datasets, this new methodology yields significant performance improvements and at the same time guarantees almost the same degree of output accuracy.

Keywords

Sequencing-by-Hybridization, Noise, Algorithmic Efficiency, Clues from the Genome

1. INTRODUCTION

DNA *sequencing-by-hybridization* (SBH) was proposed by several research groups [1, 12, 3, 13, 16, 20, 14] around 1988 as a potential alternative to wet-lab technologies. In 1991, Strezoska et al. successfully sequenced a 100bp DNA sample. To reconstruct a DNA sequence by SBH, two steps are needed: *biochemical* and *combinatorial*. At first, the target DNA sequence is brought in contact with a microarray chip of *short-length* nucleotide sequences (*probes*), so that the whole subset of probes binding to the target, called its sequence *spectrum*, is determined by some biomedical measurements. Then during the second stage, a combinatorial algorithm is applied on the spectrum to rebuild the whole sequence.

Challenges for SBH-based sequencing methods reside in both steps, we shall look at the combinatorial one in the first place. Due to the limitation of microarray technologies, researchers have been quite interested in the question that *given a small probe set, what is the maximum length of a DNA string which can be reconstructed.* [16, 15, 20] observed that by using length-k oligonucleotide probes, the expected length of unambiguously reconstructible sequences is $O(2^k)$, a bound of the same order was also proven in [6]. From $O(2^k)$ to the information theoretical bound $O(4^k)$, a real breakthrough was brought out by [18, 19], adopting *gapped* (*universal*) *bases* in probe designs. For state-of-the-art environments ($k = 8$), strategy in [19] can reconstruct several thousand bps, compared to several hundred bps previously studied.

All above analysis is based on two fundamental assumptions about the biochemical nature of DNA sequences:

1. The DNA sequences under examination are drawn from an *independently identical distribution* (*iid*) of symbols, which acts as the basic mathematical model for theoretical reasonings.
2. Sequence spectrum is perfect, no false positives or false negatives exist.

However, in reality, perfect experimental condition is only an ideal situation, especially after gapped probes are introduced which further complicates the biochemical process. This oversimplified assumption needs to be reexamined. In the past, [4, 16, 10, 15] included some redundancy in the probing scheme for error control. More recently, [2, 9] adopted a formalized random process generating *false positives* and *false negatives* in the following way.

1. Any spectrum probe can be suppressed with a fixed probability ϵ_1 (false negatives).
2. Any probe at Hamming distance 1 from a correct probe can be added to the spectrum with fixed probability ϵ_2 (false positives).

[*] The work was supported in part by the U.S. National Science Foundation NSF IIS-02-09199/IIS-03-08215. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

3. Hybridization noise is expressed in terms of error rates ϵ_1 and ϵ_2.

Under this model, Doi and Imai [2], based mainly on experimental studies, reached at a very negative conclusion that the performance of algorithms suggested in [18, 19] drops dramatically when errors exist. In a following work, Leong et al. [9] argued that the poor performance of [2] was caused by an inadequate recovery of false negatives. They showed that proper adaptation of the reconstruction algorithm leads to an acceptably graceful performance degradation. However, Leong et al. also admitted that the goal of preserving reconstruction effectiveness is achieved at a very high price of algorithmic efficiency.

Recently, another effort [17] on the theoretical frontier of SBH was published, proposing a seemingly more appropriate analog signal model based on thermodynamics of the hybridization process. Viewpoint of [17] is set from a different angle. The author discriminates between $\{A, T\}$ (weak bases) and $\{G, C\}$ (strong bases), then an *analog spectrum* of the target sequence is captured by comparing measurements with differentiated thresholds suggested by thermodynamic mechanisms, which ends up with a decreased number of false positives. This is in strong contrast to all previous studies adopting a *digital spectrum* model, where a uniform cut-off value is applied and less information is made available to the combinatorial step.

As we will explain in following sections, false negatives are much more detrimental to the reconstruction process than false positives. To cope with these probes not seen in the spectrum, lots of computing resources need to be devoted. Authors of both [9] and [17] realized this, they pointed out in their papers that, algorithmic inefficiency is a serious problem for SBH methods dealing with hybridization errors, especially false negatives. However, lab measurements are inevitable to generate something wrong.

In this paper, we reexamine this topic under the same random error model used in [2, 9], and then propose a novel algorithm which is able to greatly cut down such inefficiencies. As demonstrated by extensive experimentations, our technique yields very significant performance improvements compared to previous methods successfully taking care of errors, e.g., the one suggested in [9].

The rest of the paper is organized as follows. Section 2 briefly reviews probing schemes and reconstruction algorithms in previous works, considering both cases, i.e., with and without errors. Section 3 describes our method for sequence reconstruction, which is able to achieve much higher efficiency and in the meantime guarantee almost the same degree of output accuracy. Section 4 presents experimental results done on real DNA sequences extracted from the human genome. Section 5 concludes the paper.

2. PRELIMINARIES

Definition 1. A universal base $*$ [11] serves as a "wildcard" in subsequence matching. Two strings of identical length *coincide* if they agree on positions which are not $*$.

Definition 2. A *probing pattern* is a binary string $(0|1)^\nu$ that begins and ends with 1, where 1 denotes the position of a natural base (can be substituted by any one in $\{A, T, G, C\}$), and 0 denotes the position of a universal base $(*)$.

Definition 3. An (s, r)-*probing scheme* with probe length $\nu = (r+1)s$ uses probing pattern $1^s(0^{s-1}1)^r$. The *spectrum* of a given target sequence S_T under this scheme is the collection of all probes conforming to the chosen probing pattern that can find exact matches in S_T.

A reconstruction algorithm step-wise constructs a *putative sequence* p by adding one base to the end of p at a time, the reconstruction process is successful only if the completed putative sequence exactly coincides with the target sequence. This sequencing procedure outlined above requires some "seed" symbols to serve as contexts when "bootstrapping" and "finalizing" the task, some methods to get such "seeds" are given elsewhere in [18].

Definition 4. A probe is said to be a *feasible extension* of putative sequence p if its $(\nu - 1)$-prefix coincides with the corresponding suffix of p.

The reconstruction algorithm of [19] works in the error-free scenario. In the following of this paper, we shall refer to it as **BASIC**.

1. With putative sequence p up to the position $(i-1)$ of the target sequence, run a spectrum query to get the set of feasible extensions. Note that this set cannot be empty when p is right, since no false negatives exist.

2. If only one probe is returned, goto 3, else goto 4.

3. *Extension mode.* Add last symbol of the returned probe to the end of p.

4. *Branch mode.* Now we have an ambiguous point where all the possible branches need to be searched deeper. Considering in a graph-theoretic way, extension mode in step 3 deterministically nurtures one single path; whereas branch mode here spawns two or more competing paths simultaneously. Subsequently, based on feasible extensions, a *breadth-first* style growth of all paths is performed: for one path, if there is no way to further extend it, simply kill it. The construction of such a path tree rooted at the position $(i-1)$ is pursued up to a maximum depth H, whose meaning we will make clear soon, unless somewhere in the middle it is detected that all surviving paths have a common prefix. In the latter case, this prefix is added to the end of p, and whole procedure is switched back to extension mode where iterations continue (i.e. goto 3); otherwise, reconstruction fails, as ambiguity cannot be solved within H steps.

Definition 5. A *fooling probe* is a feasible extension whose last symbol corresponds to position i of the target sequence S_T, but its existence in the spectrum is caused by a length-ν subsequence ending at somewhere else j $(j \neq i)$ in S_T.

Fooling probes are the causes of reconstruction failures. With them, the sole authentic path cannot be distinguished from other spurious paths. However, things are not that bad. In [19], based on iid DNA sequence model, it was proven that the probability of a spurious path p_{spu}, whose presence relies on a strand of consecutive fooling probes, decreases fast when the path grows longer. Thus, what we need to do is setting an appropriately large H to make p_{spu} negligibly small (generally speaking, 2ν should be enough). Of course, the above reasonings are based on the condition that target sequence is not too long so that its spectrum is also not densely populated. In fact, this density is just the controlling factor deciding the maximum length of a DNA sequence BASIC can reconstruct with high possibility [17]. For details of an extensive analysis of fooling probes, we refer interested readers to [19, 8].

The method of Leong et al. [9] is indeed a small variation of BASIC that always operates in the branch mode: due to its breadth-first nature, we will call the algorithm as **BF** hereafter.

Before examining full details of BF, we will spend some time pondering on the different roles false positives and false negatives play in reconstruction, as promised. Going back to BASIC, a false negative directly ends the extension of authentic path, while a false positive only adds one entry to the spectrum whose effect is as same as a fooling probe. With a reasonably large H and a not so large ϵ_2, the presence of some false positives is really not a big matter.

This property was clearly realized in [2, 9] and reexamined more carefully by [17]. In BF, only false negatives are considered:

1. All four extensions (A, T, G, C) are considered possible. If one symbol is not reflected as feasible in the spectrum, add 1 to the *error score* of the corresponding path; otherwise, the score holds without change.

2. The algorithm always works in branch mode. At each step, check the error score of each path: one path is killed only if its error score is at least θ more than that of the path which has a smallest error score among all surviving paths.

Clearly, $\theta = 1$ corresponds to the ordinary path-pruning strategy depicted in BASIC. If θ is set to be greater than 1, things will be quite different. In the probabilistic sense, if false negatives are distributed in a relatively uniform way, i.e., lack of fooling probes for spurious paths is more likely than the occurrence of closely spaced false negatives on the sole authentic path [9], then it is very probable that the initial part of target sequence will be among those paths with a smallest error score (corresponding to least number of false negatives caused by extending paths in such specified ways). Readers interested in full details of a formal proof are referred to the appendix of the original paper.

The good reconstruction performance of [9] is based on the choice of $\theta = 2$, which represents a sufficient recovery of false negatives, compared to [2]. More accurate reconstruction can be achieved by setting a larger θ. However, this will lead to even more computing resource consumptions, and now BF is already quite slow.

In below, we will follow the model used in [9] with $\epsilon_2 = 0$, and propose a novel algorithm which greatly cut down BF's inefficiencies almost without sacrificing any degree of output accuracy. In the same way, we believe that inclusion of false positives should only minutely complicates the analysis.

3. FAST RECONSTRUCTION OF DNA SE QUENCES

Before describing our methodology, let us first take a closer look at BF and try to dig out the origin of its inefficiency. For the purpose of safe and adequate recovery of false negatives, BF maintains the following two types of fairness.

In the horizontal (breadth) direction, *fairness in choosing different paths for further extensions*: when reconstruction is up to one position of the target, all paths are equally extended by one additional symbol, without precedence of trying one choice before another. This strategy is safe but intrinsically very slow, because it treats every surviving path as equal and extensively searches all the possibilities, though there are some paths that are much more likely (hinting us to look at them first) whereas some other paths nearly impossible (which can somehow be pruned out).

In the vertical (depth) direction, *fairness in treating every position of the target sequence*: no matter what spot we are at during the whole procedure, a four-way ambiguity (A, T, G, C) is always assumed. Indeed, this design is necessary because false negatives exist: somewhere in the authentic path, we need seemingly "impossible" symbols (probe not seen in the spectrum) to make a further extension. However, errors are not that prevalent: For instance, if the false negative rate ϵ_1 is 0.01, then on average there will be only 1 error per $\frac{1}{0.01} = 100$ probes. Errors are very rare events; and even in the situation of encountering errors, we can often get through by using a more efficient strategy (details will be expanded later). BF assumes that completely checking all four choices at all positions is the only way to get out, but in fact this is only true for a very limited portion of all cases.

In the following, we will try to propose a different framework which successfully addresses the limitations of BF in the above two aspects. Our main idea is that, although BF can guarantee high reconstruction accuracy, in most circumstances it is too conservative and inefficient. According to our approach, we abandon fairness and make differentiations, based on this, the most suitable way is selected for reaction.

In the horizontal direction, our algorithm characterizes the differences among four extensions and utilizes an aggressive branch pruning technique based on the consideration that few errors exist (interestingly, this is a hint by observing on the vertical direction). After all these have been done, the search space becomes much smaller and is subsequently explored with the guidance of a high-probability-first heuristic, which further boosts the efficiency. Due to the bold and depth-first property of this method, we name it as **BDF**.

BDF runs fast; however, the search space BDF enumerates may potentially be too small to include the sole authentic path. There are two major reasons for it. Apart from the one discussed above (indeed a necessary price paid for being adventurous), there is another point which is related to guaranteeing BDF's correctness. The latter will be the main focus of subsection 3.3.

Considering all the above factors, one needs to come up with a good solution. As what has been mentioned, BF treats every position of the target in the same and most primitive way, i.e., simply obtains the authentic path by greatly enlarging the search space.

In comparison to this naive scheme, our algorithm again tries to abandon fairness on the vertical direction. It decides different regions during reconstruction and then applies a divide-and-conquer paradigm.

1. Rush at full speed whenever possible (BDF).

2. Make necessary slow-downs and be meticulous while sequencing some specific subparts of the target (BF).

As shown by comprehensive experimental results, case 1 is a dominant existence, and substantial acceleration can be achieved.

3.1 Probabilistic differences among four ex tensions

Different paths are of varying probabilistic strengths. Naturally, such bias is a very useful guide when ambiguities are encountered during reconstruction. Compared to the breadth-first style BF, BDF has a nice feature that, if along one path the target is successfully sequenced, then we do not need to go back to the branching point and consider other remaining paths. In principle, the bigger the bias is, the more effective BDF will be.

We now turn to a more formal modelling of the notion brought out above. Previously, information from the biomedical step is used in an obvious and minimal way: presence of one probe in the spectrum is treated as individual units, without combining any neighbor (other probes) or background (classification of the target sequence with regard to different categories of genomes) information, which are very useful clues.

For ease of illustration, we assume a $(4,4)$-probing scheme in the remainder of the paper. This assumption is a good representation for the state-of-art microarray technology and has been frequently considered in many previous studies [18, 19, 2, 8]. However, our methodology is not confined to this narrow situation.

Let us first look at the interactions existing among multiple probes in the spectrum. As Figure 1 depicts, when the symbol at position i needs to be decided, i.e., the putative sequence is up to the position $i-1$, we can align 5 probes $\{b_0, b_1, \cdots, b_4\}$ in the shown manner, with b_j ending at position $i_j = i + 4j$ $(j = 0, 1, \cdots, 4)$.

Denote the symbol at position i to be s_i, then given a specific setting of $(s_{i_0}, s_{i_1}, s_{i_2}, s_{i_3}, s_{i_4})$, all five probes $\{b_0, b_1, b_2, b_3, b_4\}$ must be included in an error-free spectrum, since they are necessary proofs for such an assignment of symbols.

Definition 6. If the spectrum contains five probes $\{b_0, b_1, b_2, b_3, b_4\}$ whose mutual alignment is the same as what Figure 1 depicts, then $(s_{i_0}, s_{i_1}, s_{i_2}, s_{i_3}, s_{i_4})$ is called a *length-*5 *feasible extension.*

When false negatives are not considered, things are relatively straightforward:

$$\begin{aligned} & P(s_{i_0} = x_0) \\ = & \sum_{x_1, \cdots, x_4} P(s_{i_0} = x_0, s_{i_1} = x_1, \cdots, s_{i_4} = x_4) \end{aligned}$$

Ignoring the situation that different length-5 feasible extensions may have different *a prior* probabilities (we will examine it later), and let

$$\begin{aligned} & P(x_0, x_1, \cdots, x_4) \\ = & P(s_{i_0} = x_0, s_{i_1} = x_1, \cdots, s_{i_4} = x_4) \end{aligned}$$

we have:

$$P(x_0, x_1, \cdots, x_4) = \frac{1}{N}$$

if $(x_0, x_1, \cdots, x_4)$ is a length-5 feasible extension, whereas N is the total number of length-5 feasible extensions; and

$$P(x_0, x_1, \cdots, x_4) = 0$$

otherwise.

Figure 2 shows a small example of above formula. Under the circumstance depicted there, we have $P(s_i = A) = \frac{3}{4}$ and $P(s_i = G) = \frac{1}{4}$. What is better, based on our experiences, we may have $P(s_i = x) = 1$ in many cases, which means ambiguity can sometimes be solved by aligning more than one probes together. In [19], this nice property was also noticed.

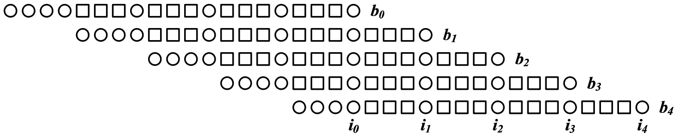


Figure 1: Switch 4 positions right and count number of paths: circle – natural base, square – universal base; here, $\{b_0, b_1, b_2, b_3, b_4\}$ must coincide on those "circles", while "squares" are indeed "don't cares"

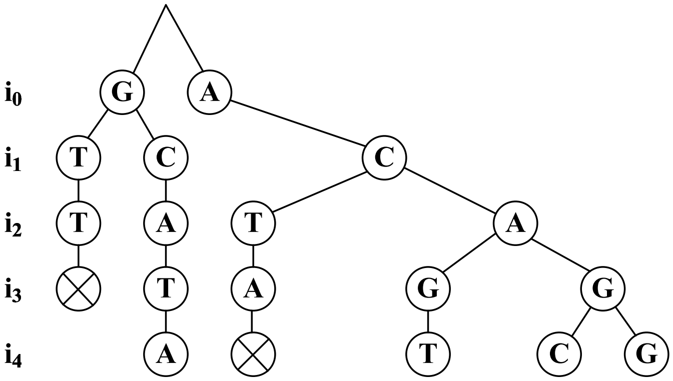


Figure 2: An example tree showing feasible length-5 extensions: cross – failure to find a probe in the spectrum corresponding to that position

Some minor revisions are needed to treat false negatives. In Figure 2, false negatives can possibly occur at a position where one cross is placed. As a remedy here, now the corresponding branch will not be ended anymore, i.e., all length-5 extensions are considered to be feasible. But as we have described earlier, false negatives are very rare events. To formally model this concept, we have: for one specific assignment of $(s_{i_0}, s_{i_1}, s_{i_2}, s_{i_3}, s_{i_4})$, if $f (f \leq 5)$ probes (among $\{b_0, b_1, b_2, b_3, b_4\}$) necessary to support its validity are missing from the spectrum, then e^f is considered to be a strength (or weight) w of the length-5 extension when summing up probabilities in the formula, and

$$e = \frac{\epsilon_1 \times \text{target sequence length}}{4^8}$$

denotes an approximation of the probability that one probe not present in spectrum is indeed a real false negative.

After integrating the strength of different length-5 extensions, the probability calculation formula becomes:

$$\begin{aligned} & P(s_{i_0} = x_0) \\ = & \frac{\sum_{x_1, \cdots, x_4} w_{x_0, x_1, \cdots, w_4} P(x_0, x_1, \cdots, x_4)}{\sum_{x_0, x_1, \cdots, x_4} w_{x_0, x_1, \cdots, w_4} P(x_0, x_1, \cdots, x_4)} \end{aligned}$$

3.2 Clues from the genome

Besides what has been elucidated, there is another important observation leading to different modelling of length-5 extensions' strengths: a species' genome is not uniform,

P	μ(mean)	σ(std)
A	0.282	0.096
T	0.288	0.086
G	0.212	0.121
C	0.218	0.085

Table 1: Human Chromosome 7 is found to be not uniform

as opposed to the first assumption mentioned in section 1, which is taken for granted in many previous works.

Taking Human Genome Chromosome 7 for instance (approximately 150 million bases long), we simply calculate the association between one symbol and seven symbols preceding it, i.e., $P(s_i|s_{i-1}s_{i-2}\cdots s_{i-7})$, which is in fact a relatively simple Markov DNA sequence model [7]. Below is the result we found:

1. we calculate $P(s_i|s_{i-1}s_{i-2}\cdots s_{i-7})$ for all 16384 possibilities of $s_{i-1}s_{i-2}\cdots s_{i-7}$, then get averages and standard deviations, which is shown in Table 1.

2. If we set $t = 0.4$ to be a threshold level, then among all 16384 different cases, there are 3982 with predicted probability of one particular symbol higher than t. It is better to think like this: say the symbol is A, then A's probability is on average a double of T (or G, C)'s probability, i.e., $0.4 = 2 \times \frac{1-0.4}{3}$, and the usefulness of such a bias should be non-trivial in analysis.

However, during experimentation, we did not observe any significant improvement by integrating the background information in such a primitive way. One possible reason is that, in noisy situations, the bias introduced here is less visible when compared with the ones mentioned in 3.1. Nevertheless, using clues from the background is still a major motivation for us to do this work, and thus we choose to use real human genome dataset instead of synthetic sequences generated by iid distributions. Our belief is that more expert knowledge on how to utilize the information more accurately, e.g., distinguishing between exons and introns and train models separately [7] or a Hidden Markov Model (HMM) for DNA sequences [5], can definitely help further promote the reconstruction algorithm's performance.

3.3 Controlling the accuracy of BDF

Talking about accuracy, we may have two kinds of errors: type I and type II. In the sequence reconstruction problem examined here, a *type I error* happens if we wrongly kill the authentic path while search space is being pruned; a *type II error* happens if the algorithm finishes and reports a spurious path. In below of this subsection, we first discuss in detail some aspects of BDF which can possibly generate type I and type II errors, and then move on to a technique fighting against type I errors, a design aiming at minimizing the number of type II errors is explained last.

In our algorithm, there are two potential sources of type I errors:

1. Termination of path extension. In case of unsatisfying path growth, i.e. too many probes necessary to support putative sequence p's validity are missing from the spectrum, BDF chooses to *backtrack* and p is pruned out. This decision can be made with regard to p's length l and ϵ_1, where mean$(e) \approx \epsilon_1 l$.

2. Branch selection. In subsection 3.1, we mainly discussed a way to rank $\{A, T, G, C\}$ according to each of them's probability, so that a high-probability-first heuristic can be utilized. Furthermore, in order to make the whole space substantially smaller, we also want to aggressively prune out some choices. These branches, though possible, are of negligible likelihoods.

   The optimization we adopt here is: for one specific assignment of $(s_{i_0}, s_{i_1}, s_{i_2}, s_{i_3}, s_{i_4})$, f (defined in 3.1) should be no more than 1, i.e. $f \leq 1$.

There are also two situations for type II errors:

1. To avoid unnecessary inefficiencies, we want to fix one ambiguous position i after successfully progressing H' steps, and BDF will not backtrack beyond i in later phases. The basic thinking here is that, if symbol i is wrongly guessed, then the probability that many fooling probes will appear to support this length-H' spurious path is very low. The same notion is behind the design of H in BASIC.

2. By "successfully progressing", we mean no backtrackings can be triggered. Thus, type II errors can also happen if the backtracking condition is too relaxed.

Then things are much clearer: If position i is wrongly fixed (type II error), there is no way for BDF to get it right later; in comparison, if we wrongly kill the authentic path (type I error) and backtrack to a fixed position, i.e., BDF cannot get through and is about to fail, BF comes as a backup. Since the search space of BF is much larger, the strategy in overall is able to fight against type I errors, at least as well as pure BF can. Different regions of reconstruction is now finally characterized. A newer version of BDF is as follows.

NBDF: while the end of target is not reached:

1. Perform depth-first search following probability precedences and aggressive pruning. If backtracking condition is not triggered, go deeper in the branch: After successfully progressing H' steps, fix a previously guessed position. Otherwise, kill the path and move backwards.

2. When the depth-first search is backtracked to a previously fixed position, instead of failure, call BF_Helper(), a variant of BF which runs for D_{BF} steps. Then the path with the least number of errors is picked out and appended to the end of current putative sequence p. D_{BF} is chosen to trade off between accuracy and efficiency: with larger D_{BF}, the algorithm is closer to pure BF, which is safe but slow. In our program, we set it to be 80 without very careful tuning.

3. Switch back to depth-first, i.e., goto 1.

NBDF's design offers the advantage that we can treat type I and type II errors in a biased way. Previously, designing a successful backtrack condition is difficult, because it is related to both error types and from hypothesis testing we know that a gap surely exists. Now because of BF's backup, we can choose to minimize the number of type II errors, i.e.,

	0.001	0.002	0.005	0.01	0.02
NBDF					
1000	4.0s	4.2s	5.3s	6.5s	33.8s
1500	9.7s	9.7s	11.3s	24.7s	60.5s
2000	13.3s	17.2s	28.9s	59.4s	175.0s
2500	16.6s	22.4s	35.0s	129.5s	402.2s
3000	42.0s	38.1s	90.1s	213.1s	576.8s
BF					
1000	35.1s	36.8s	48.7s	37.6s	57.0s
1500	144.0s	135.0s	134.0s	132.1s	151.2s
2000	578.6s	560.8s	480.4s	514.9s	471.1s
2500	1007.7s	1055.6s	958.5s	914.4s	873.4s
3000	2264.3s	1454.3s	1557.8s	1927.0s	1293.2s
NBDF/BF					
1000	11.4%	11.3%	10.9%	17.3%	59.3%
1500	6.7%	7.2%	8.4%	18.7%	40.0%
2000	2.3%	3.1%	6.0%	11.5%	37.1%
2500	1.6%	2.1%	3.7%	14.2%	46.0%
3000	1.9%	2.6%	5.8%	11.1%	44.6%

Table 2: NBDF and BF's computation time comparison w.r.t. (L, ϵ_1)

	0.001	0.002	0.005	0.01	0.02
NBDF					
1000	47	47	47	47	47
1500	45	45	45	45	42
2000	46	46	44	44	33
2500	45	44	44	38	27
3000	37	38	37	37	21
BF					
1000	48	48	49	48	46
1500	45	45	45	46	45
2000	47	47	46	45	36
2500	47	47	46	41	30
3000	39	41	37	33	21
NBDF/BF					
1000	98%	98%	96%	98%	102%*
1500	100%	100%	100%	98%	93%
2000	98%	98%	96%	98%	92%
2500	96%	94%	96%	93%	90%
3000	95%	93%	100%	112%*	100%

Table 3: NBDF and BF's reconstruction accuracy comparison w.r.t. (L, ϵ_1)

the reason for backtracking is: "we have to do it, otherwise the probability of correctly fixing one previously guessed position after H' steps will be low"; but not "current putative path p is very unlikely to be the authentic one", which aims at controlling type I errors.

According to the proof of Theorem 1 in [19], there are three error events if we wrongly set one ambiguous position after H' steps (there, BASIC's focus is on H): E_1 is due to probabilistic extensions using fooling probes, E_2 and E_3 are related to deterministic extensions caused by two identical subsequences in the target. Since only E_1 depends on the choice of H' while E_2 and E_3 reflects the data property, we will discuss E_1 in detail here. More descriptions of the rest two are not repeated and can be found elsewhere in previous literature [19, 9].

Assume that a spurious path, starting at position i, is extended to position $i + H'$. Let P_h be the probability of extending up to position $i + h$, then clearly $P_0 = 1$. If the extension from $i+h$ to $i+h+1$ is caused by a fooling probe, then the situation is the same as that in [19]:

$$P_{h+1} < P_h \times P_{fool} = P_h \times (\frac{m}{4^{k-1} + \frac{4}{3}(\frac{1}{c^r} + \frac{1}{4^{s-1}})})$$

Now we calculate the probability of an extension caused by false negative recovery, i.e., no supporting probes are seen in the spectrum. According to our aggressive way of branch selection ($f \leq 1$), a guess can be placed if and only if four additional switches are supported by probes seen in the spectrum, i.e., there will be totally 5 positions (including the guess position) specified without other choices. On the other hand, Probability P_{guess} that four additional probes support the guess is given by Lemma 2 in [19]:

$$P_{guess} = 4(\frac{m}{4^k} + \frac{1}{3 \times 4^{s-1}})^r$$

where the "4" outside parenthesis means that all four symbols (A, T, G, C) are possible.

Combining the above two, the probability that the extension can be made up to H' with at most e guesses is upper bounded by:

$$P_{H'} < (P_{fool})^{H'-5e} \times (P_{guess})^e$$

In our implementation, H' is chosen as 60 and e is set as as 2, which means that BDF will not backtrack if in 60 steps there are less than 3 positions not supported by any probe in the spectrum. Given this reasonably large H' and relatively small e, the probability of E_1, i.e., $P_{H'}$, is negligible.

4. EXPERIMENTAL EVALUATIONS

In this section, we compare our NBDF algorithm with the BF algorithm defined in [9], which is known to be a successful strategy dealing with SBH under noisy conditions, especially for false negatives. We evaluate the performance of these two algorithms using real DNA datasets downloaded from www.ensembl.org: Human Chromosome 7. All experiments were implemented by Microsoft Visual C++ 6.0 and run using a 3GHz Pentium IV machine with 1GB main memory.

4.1 Experiment settings

We mainly compare the speed and accuracy performance of NBDF and BF with regard to sequence length L and false negative rate ϵ_1, because they are two major factors governing the hardness of reconstruction problem. The same testing strategy was used in [9].

For a given choice of testing configuration (L, ϵ_1), we pick 50 contigs from the dataset, obtain each sequence's spectrum and subsequently modify it by suppressing the existence of some probes with a uniform probability of ϵ_1. Initiation and termination steps of the reconstruction, i.e., getting first and last several symbols, are separated from NBDF and BF's executions, which is reasonable because there have been some ways proposed to complete this task [18].

Now NBDF and BF can be run for all configurations. Sequencing success will be reported only when the target

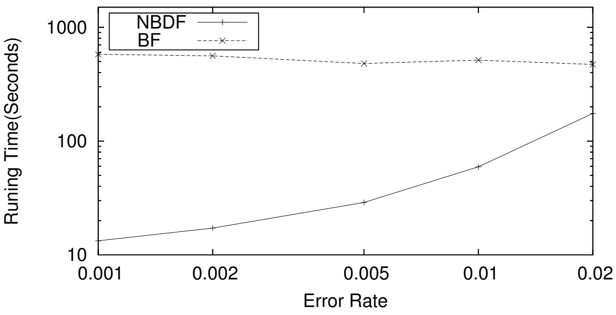


Figure 3: Computation time for reconstruction w.r.t. ϵ_1, when $L = 2000$

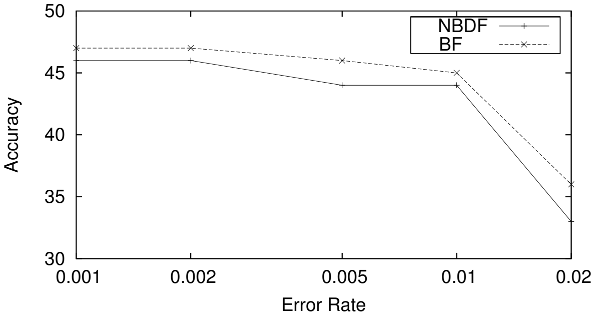


Figure 4: Reconstruction accuracy w.r.t. ϵ_1, when $L = 2000$

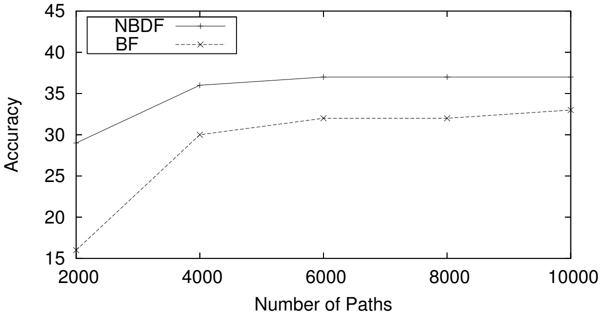


Figure 5: Reconstruction accuracy w.r.t. memory upperbound, when $L = 3000$ and $\epsilon_1 = 0.01$

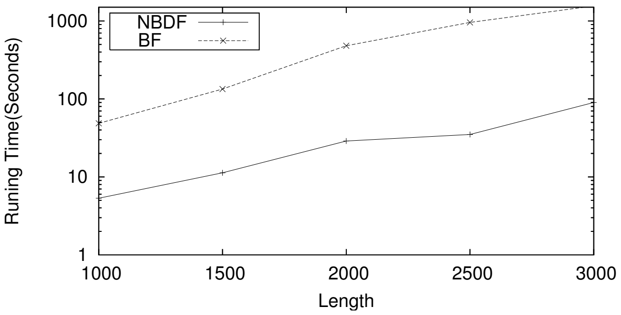


Figure 6: Computation time for reconstruction w.r.t. L, when $\epsilon_1 = 0.005$

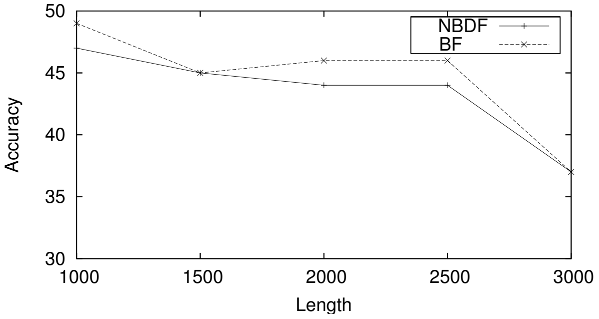


Figure 7: Reconstruction accuracy w.r.t. L, when $\epsilon_1 = 0.005$

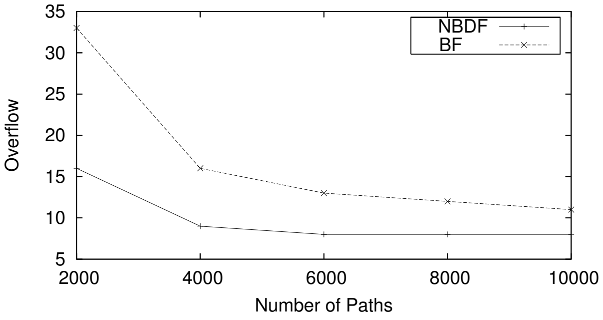


Figure 8: Number of overflows w.r.t. memory upperbound, when $L = 3000$ and $\epsilon_1 = 0.01$

sequence completely coincides with the algorithms' output.

4.2 Superiority of NBDF

The behaviors of NBDF and BF are summarized: Tables 2-3 are the complete speed and accuracy specs of these two algorithms for all testing configurations, whereas Figures 3, 4, 6 and 7 directly compare NBDF and BF's performance with regard to one aspect while the other is fixed to a "moderate" value, i.e., $L = 2000$ or $\epsilon_1 = 0.005$.

From empirical studies, two things are clear:

1. NBDF is orders of magnitude faster, especially when ϵ_1 is low and L is high. Reasons are obvious for such situations: if ϵ_1 is low, then most part of the target sequence is "easy" to reconstruct; if L is high, since BF's search breadth will in general grow wider and wider towards the end of the sequence, extending one symbol in foreground means more paths to be checked in background. Thus, BF's unnecessary waste of time is high.

2. NBDF's output is almost of the same quality as BF. For most cases, within 50 contigs, NBDF and BF's accuracy difference is one sequence or two sequences sometimes.

There are some other interesting patterns shown in the results.

1. BF's time curve w.r.t. false negative rate is relatively "flat", implying that a non-discriminative brute-force strategy is used.

2. Memory size is crucial for BF's performance. As L and ϵ_1 become larger, the general case is that BF has to keep more parallelling paths in storage. Given a fixed upperbound (in our experiments it is 10000 paths),

sooner or later it may blow up. In comparison, NBDF calls BF_Helper() only when necessary and the subroutine executes within a short range, which is a more suitable and robust way of doing things, especially when error rate is not so high and the sequence really can be reconstructed. For instance, BF got 33 sequences on ($L = 3000, \epsilon_1 = 0.01$) while NBDF got 37 (see those entries with a "*" in Table 3). Figures 5 and 8 depict such circumstances and compare two algorithms' performance with regard to a preset memory upperbound. We also tested a couple of sequences with $L = 5000$, and a same situation is observed, except that BF is even slower and overflows almost on every test case. Even if we can surely assign more memory to BF, it is not an ultimate solution because of the drastic processing time.

3. From the charts, it seems queer that BF's running time sometimes drops as the error rate increases, which is indeed natural because less time needs to be spent when BF overflows and gets out before reaching the end of target.

5. CONCLUSIONS

In this paper, we carefully examined the SBH problem under a random hybridization error model and showed with extensive analysis that the previously suggested strategy using pure breadth-first search is too simple to be efficient. It is safe but intrinsically very slow.

Then, we proposed a different divide-and-conquer technique NBDF which calls safe BF (a variant of the previous algorithm) to solve those "hard" portions and fast BDF to tackle the rest. Due to BDF's depth-first nature, we need to define a good backtracking condition so that both type I and type II errors can be avoided with high possibility. The hypothesis testing gap between these two types is finally solved by focusing on type II, for which we gave a theoretical bound of the error probability. And when type I errors indeed happen, a "hard" region in front is noticed, where BF is executed to exert its ability of searching a much larger space (though slower).

Finally, we conducted a comprehensive empirical study on real human DNA sequences. It turned out that NBDF is orders of magnitude faster, while the output accuracy is also guaranteed. Another notable advantage of our technique compared to the previous one is that it does not necessarily require a very large main memory size to avoid potential overflows.

6. REFERENCES

[1] W. Bains and G. Smith. A novel method for DNA sequence determination. *Journal of Theoretical Biology*, 135:303–307, 1988.

[2] K. Doi and H. Imai. Sequencing by hybridization in the presence of hybridization errors. *Genome Informatics*, 11:53–62, 2000.

[3] R. Drmanac, I. Labat, I. Bruckner, and R. Crkvenjakov. Sequencing of megabase plus DNA by hybridization. *Genomics*, 4:114–128, 1989.

[4] R. Drmanac, I. Labat, and R. Crkvenjakov. An algorithm for the DNA sequence generation from k-tuple word contents of the minimal number of random fragments. *Journ. Biomolecular Structure and Dynamics*, 8:1085–1102, 1991.

[5] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids.* Cambridge University Press, 1998.

[6] M. Dyer, A. Frieze, and S.Suen. The probability of unique solutions of sequencing by hybridization. *Journal of Computational Biology*, 1:105–110, 1994.

[7] W. Ewens and G. Grant. *Statistical Methods in Bioinformatics: An Introduction.* Springer-Verlag, 2001.

[8] S. Heath and F. Preparata. Enhanced sequence reconstruction with DNA microarray application. *COCOON*, pages 64–74, 2001.

[9] H.-W. Leong, F. Preparata, W.-K. Sung, and H. Willy. On the control of hybridization noise in DNA sequencing-by-hybridization. *WABI*, pages 392–403, 2002.

[10] R. Lipshutz. Likelihood DNA sequencing by hybridization. *Journ. Biomolecular Structure and Dynamics*, 11:637–653, 1993.

[11] D. Loakes and D. Brown. 5-nitroindole as a universal base analogue. *Nucleic Acids Research*, 22(20):4039–4043, 1994.

[12] Y. Lysov, V. Florentiev, A. Khorlin, K. Khrapko, V. Shih, and A. Mirzabekov. Sequencing by hybridization via oligonucleotides, a novel method. *Nucleic Acids Research*, 303:1508–1511, Dokl. Acad. Sci. USSR.

[13] P. Pevzner. l-tuple DNA sequencing: computer analysis. *Journ. Biomolecular Structure and Dynamics*, 7(1):63–73, 1989.

[14] P. Pevzner. *Computational Molecular Biology: An Algorithmic Approach.* MIT Press, 2000.

[15] P. Pevzner and R. Lipshutz. Towards DNA-sequencing by hybridization. *19th Symp. on Mathem. Found. of Comp. Sci.*, pages 143–158, 1994.

[16] P. Pevzner, Y. Lysov, K. Khrapko, A. Belyavsky, V. Florentiev, and A. Mirzabekov. Improved chips for sequencing by hybridization. *Journ. Biomolecular Structure and Dynamics*, 9(2):399–410, 1991.

[17] F. Preparata. Sequencing by hybridization revisited: the analog-spectrum proposal. *IEEE Transactions on Computational Biology and Bioinformatics*, 1(1), January 2004.

[18] F. Preparata, A. Frieze, and E. Upfal. On the power of universal bases in sequencing by hybridization. *Third Annual International Conference on Computational Molecular Biology*, pages 295–301, April 1999.

[19] F. Preparata and E. Upfal. Sequencing-by-hybridization at the information-theory bound: an optimal algorithm. *Proceedings of the fourth annual international conference on Computational molecular biology*, pages 245–253, 2000.

[20] M. Waterman. *Introduction to Computational Biology.* Chapman and Hall, 1995.

On Discovery of Maximal Confident Rules without Support Pruning in Microarray Data.

Tara McIntosh*
School of Information Technology
The University of Sydney
Sydney, Australia
tara@it.usyd.edu.au

Sanjay Chawla
School of Information Technology
The University of Sydney
Sydney, Australia
chawla@it.usyd.edu.au

ABSTRACT

Microarray data provides a perfect riposte to the original assumption underlying association rule mining – large but sparse transaction sets. In a typical microarray the number of columns (genes) is an order of magnitude larger than the number of rows (experiments). A new family of row enumerated rule mining algorithms have emerged to facilitate mining in dense sets. However, to date, all the algorithms proposed to mine expression relationships alone rely on the support measure to prune the search space. This is a major shortcoming as it results in the pruning of many potentially interesting rules which have low support but high confidence. In this paper we propose the MaxConf algorithm which exploits the weak downward closure of confidence to directly mine for high confidence rules. We also provide a means to evaluate the biological significance of the gene relationships identified. An evaluation of MaxConf with RERII on the database BIND shows that their *recall* is 94% and .15% respectively.

Keywords

Microarray, Association Rules, Row-Enumeration, Maximum Confidence

1. INTRODUCTION

The increasing volume of biological data collected in recent years has prompted considerable interest in developing advanced and efficient bioinformatic tools for genomic and proteomic data analysis. The *microarray* is considered revolutionary in the biological domain as it allows one to study the behaviour of all the genes within a cell in only one experiment. One main objective of biologists is to develop a deeper understanding of how cells regulate gene expression and other cellular tasks. These mechanisms can be depicted in gene networks. However still today most genes known to be involved in a particular process are identified in painstaking molecular and genetic wet-lab experiments which allow only a few genes to be studied at a time. This is mainly due to the large volumes of data that microarray experiments return to biologist who have very limited ways for extracting useful information.

Association rule mining is a foundational technique which allows for the simultaneous discovery of potentially interesting relationships. Mining algorithms have the potential to extract interesting patterns from microarray expression data which may aid in the identification of gene networks, where the expression of a gene can depend on the expression of others:

$$\text{Gene1} \Rightarrow \text{Gene2 (support 10\%, confidence 90\%)}$$

The above rule states that when Gene1 is expressed 90% of the time Gene2 is also expressed, and Gene1 and Gene2 are expressed together in 10% of the experiments.

Association rule mining has been shown to be very effective for analysing microarray data. For instance, Creighton and Hanash [9] applied Apriori rule mining [1] to the popular Hughes et al. [13] microarray dataset of *S.cerevisiae.* Many of the rules generated were consistent with biological knowledge, and other rules revealed numerous unexpected relationships that warranted further biological investigation. The associations generated revealed correlations between many genes that were not identified from clustering methods [9].

When datasets consist of a large set of items and far less transactions as is the case for microarrays, Apriori style algorithms suffer from itemset explosion, often rendering them inappropriate for data analysis. Both [15] and [17] showed that by searching the *row enumeration* space, the complete set of frequent closed itemsets can be obtained avoiding itemset explosion. Compared to *item enumeration* methods like Apriori, row enumeration is a top-down approach starting with each transaction being a candidate itemset and iteratively removing items to form smaller candidates of greater support. Cong et al. [6] presented RERII, an efficient row enumeration based algorithm best suited to datasets where the number of items greatly outweighs the number of transactions.

Recently row enumeration methods like FARMER have been developed with the intent to construct rule based classifiers [8, 7]. Such classification methods differ to the one we

*Work done in part of undergraduate honours project

BIOKDD '05, August 2005, Chicago, Illinois, USA
Copyright 2005 ACM 1-59593-213-5 ...$5.00.

propose here (MaxConf) in that they require a predetermined class for each experiment which becomes the consequent of the rules. MaxConf on the other hand can generate rules from unclassified microarrays, with no consequent restriction. MaxConf was designed with the intention to analyse perturbation microarray data to aid the construction of gene networks on a global genome scale, compared to previous algorithms such as Boolean Networks which are restricted to the number of genes that can be incorporated into the analysis [2, 3].

However there is a fundamental issue related to the limitation of support-based pruning that previous algorithms do not address. Namely, many rules that a biologist would consider of high interest are pruned (because of support pruning), leaving them undiscovered. One of the main objectives of this paper is to propose a technique to lift this limitation, to aid in future gene network construction.

Despite the rapid introduction of many association rule algorithms to analyse microarray data, little research has been directed to the validation of the resulting rule sets. For this reason, most analyses have been directed to performance studies with respect to time and space requirements. For example [6] restricted their analysis to performance comparisons with state of the art Apriori style methods, CHARM [19] and CLOSET [16]. This forms the basis for our second objective; to introduce and encourage the use of publically available biological databases to validate relationships. We take on this approach to further evaluate our MaxConf algorithm.

1.1 Main Contributions

1. A confidence based top-down algorithm MaxConf for identifying interesting gene relationships on a global scale is proposed and implemented. This algorithm efficiently identifies high confident rules without support pruning achieving significant recall improvements.

2. We have designed a systematic framework to validate the rules discovered using two highly regarded biological databases, *BIND*[5] and the *Gene Ontology* [18]. These databases allow us to compare our results with previous methods with respect to recall, precision and biological significance. On the *BIND* database the recall of MaxConf and RERII is 94% and 0.15% respectively.

1.2 Preliminary

When applying association rule mining to a microarray dataset D, the set of items I refers to the set of genes studied on the microarray, and each transaction t corresponds to an individual experiment. An association rule R of the form $I_1 \Rightarrow I_2$, where $I_1, I_2 \subseteq I$ may be generated from D. As defined in [1], the antecedent and consequent of R correspond to I_1 and I_2, which we denote by ante(R) and cons(R) respectively. Further we represent the support of an itemset I as supp(I) and the confidence of the rule R as conf(R).

The rest of this paper is as follows. In Section 2 we will briefly introduce an association rule mining method designed to specifically mine high confident rules. In Section 3 we build the core machinery for directly discovering high confidence rules. A series of definitions and their elementary properties will culminate in Algorithm 1 - the MaxConf algorithm. In Section 4 we will compare the MaxConf algorithm with RERII in three dimensions - running time for different combinations of support, confidence and number of rules generated. In Section 5 we detail our approach to analyse the biological significance of the gene relationships (rules) we discover. We conclude in Section 6 with a summary and directions for future work.

2. HIGH CONFIDENCE RULES

In this section we introduce a previous approach (Apriori-MaxPI) designed to address the fundamental shortcoming of support based mining.

2.1 Apriori MaxPI

The support based techniques deem infrequent itemsets unfavourable (unsupported), resulting in them being pruned during frequent itemset generation. Therefore in the following iteration, only a subset of confident rules will be mined. However it is often the high confidence rules that occur with low frequency which present interesting characteristics within the dataset.

The *Maximal Participation Index* (maxPI) was introduced in [12] to mine co-location patterns from spatial datasets. It excludes the support threshold from the search, allowing all confident rules to be identified.

Definition 1 (Maximal Participation Index). *Given an itemset I the maximal participation index of I is defined as the* maximal participation ratio *(pr) of all items $i \in I$.*

$$\begin{aligned} maxPI(I) &= max_{i \in I}\{\, pr(I,i)\,\} \text{ where} \\ pr(I,i) &= conf(i \Rightarrow (I/i)) \\ &= \frac{supp(I)}{supp(i)} \end{aligned}$$

From Definition 1 it is clear that the maxPI of an itemset is the maximum confidence a generated rule can have. If the maxPI of an itemset is below the confidence threshold it cannot generate any confident rules. Unlike support, maxPI is not monotonic with respect to itemset containment relations. However, maxPI does exhibit a *weak monotonic property* (Definition 2). Applying this property an Apriori style algorithm (which we call Apriori-maxPI) to mine confident itemsets is possible. Based on this property, if a k-itemset is maxPI frequent, then at most one of its subsets with (k-1) items is not confident.

Definition 2 (maxPI weak monotonicity). *Let I_1 be a k-itemset. Then there exists at most one (k-1) subsets I_2 where $I_2 \subset I_1$ such that maxPI(I_2) < maxPI(I_1).*

One drawback of using maxPI is that no single itemsets can be pruned in the first phase of Apriori as they all have a confidence of 100%. Therefore Apriori-maxPI must deal with all the singleton candidate itemsets and the $|\mathcal{I}|^2$ 2-itemsets. It is not until 3-itemset candidates are generated that pruning can be applied. Based on support alone, if any (k-1)-itemset of a k-itemset is not frequent, then the k-itemset can not be frequent, and thus can be pruned without a need to do support counting. With respect to maxPI, a k-itemset is only guaranteed to not be maxPI frequent (maxPI

$\geq$ minimum confidence) if more than k-2 (k-1)-subsets are not. Therefore maxPI pruning is not as stringent as that using support.

This property works against Apriori, which works efficiently on the assumption that the number of frequent itemsets is low. Further with a large number of items in microarray data, Apriori-maxPI approaches suffer from itemset explosion.

Unfortunately there is no property of maxPI that can be exploited by a top-down approach, without potentially losing confident rules. Motivated by this issue, we have identified a property of confidence that can be utilised by a top-down algorithm which we describe in the following section.

3. MAXCONF

The main challenge in devising a top-down algorithm to mine high confidence rules is that no support pruning can take place. A naïve approach in a top-down manner would be to grow the entire row enumeration tree until no itemsets can be generated. This would be equivalent to generating all closed itemsets (including those that are not frequent with respect to support). From these all confident rules may be generated. Concerning microarrays (and other dense datasets), the set of closed itemsets is already extremely large, many of which cannot generate confident rules and as such the naïve approach requires unnecessary expensive computations and memory. We applied this naïve approach, which reported an error after using up all available memory, when only 30% of the transactions had been processed.

In this section we introduce our top-down approach to mining *maximal confident rules* efficiently. Our algorithm MAXCONF (Algorithm 1) addresses the main shortcomings of association rule mining. MAXCONF exploits two pruning methods each based on confidence allowing us to prune the row-enumeration tree without losing any rules. It is further enhanced to only mine all *maximal confident rules*, reducing the final rule set size inline. Each of these methods are explained in the following subsections.

3.1 Level 1 Confidence Pruning

This pruning is based on an observation of the structure of the row enumeration tree and is performed on line 11. At any point in the row enumeration tree we can predict the maximum support [6] and confidence an itemset can exhibit, based on its location within the tree. From this property our first pruning technique is possible, which is detailed in the following definitions.

DEFINITION 3 (MAXIMUM SUPPORT). *Given a node N with k child nodes, $N_1, \dots, N_k$, for any child node N_i the* maximum support *of N_i or any of its potential child nodes is:*

$$maximum\ supp \quad = \quad N_i.initial\_supp + k - i$$

DEFINITION 4 (MINIMUM FEATURE). *Given an itemset I, the item $i_1 \in I$ is the* minimum feature *if:*

$$minimum\ feature \quad = \quad supp(i_1) \leq supp(i_2) \mid \forall i_2 \in I$$

DEFINITION 5 (SPANNING RULE). *Given an itemset I, a rule r spans I if*

$$ante(r) \cup cons(r) = I$$

$$|ante(r)| = 1$$

Algorithm 1 MaxConf - maximal confident rule mining

```
1: for all transactions $t \in \mathcal{D}$ do
2:   // No single item set pruning
3:   N := $\emptyset$
4:   n := new Node(items = t.items, support = 1)
5:   N.append(n)
6: MR := $\emptyset$ // set of maximal confident rules
7: MaxConf_depthfirst(N)
8: Procedure: MaxConf_depthfirst(N)
9: for all node $n_i \in N$ do
10:   Children := $\emptyset$
11:   if $n_i$ cannot be confident then
12:     continue // Level 1 pruning
13:   Determine support of $n_i$, populate Children, and
      prune based on closure only as in RERII.
14:   M:= getMaxFeatures($n_i$) // Line 27
15:   if M $\neq \emptyset$ then
16:     for all $m \in M$ do
17:       if $m \notin n_i$.maxFeatures then
18:         MR.append($m \Rightarrow \{n_i.items \setminus m\}$)
19:     for all child $c \in Children$ do
20:       if $c \subset$ M then
21:         delete $c$ // Level 2 pruning
22:       else
23:         c.maxFeatures.insert($n_i$.maxFeatures)
24:         c.maxFeatures.insert($c \cup$ M)
25:   if Childen $\neq \emptyset$ then
26:     MaxConf(Children)
27: Procedure: getMaxFeatures(n)
28: M := $\emptyset$ // set of maximal features
29: for all items $i \in n.items$ do
30:   if support($n$) / support($i$) $\geq$ min_confidence then
31:     M.insert($i$)
32: return M
```

DEFINITION 6 (MAXIMUM CONFIDENCE). *Given a node N, let σ be the maximum support of N and i be the minimum feature of N. The maximum confidence of any spanning rule of N is:*

$$maximum\ confidence \quad = \quad \frac{\sigma}{supp(i)}$$

If we know that the maximum confidence of a node's itemset (i.e. maximum confidence of the rule which spans the node) is less than the confidence threshold, it can be pruned, as any further enumeration below the node will only generate less or equally confident child itemsets.

3.2 Level 2 Confidence Pruning

We identified the *weak downward closure* property of the confidence measure, which we can exploit during the enumeration tree generation process, to effectively prune nodes which will provide no new information. This pruning is performed on lines 19-21 and is based on the following definitions:

DEFINITION 7 (MAX FEATURES). *Given an itemset I, let R_I be the set of all confident rules $\{x \Rightarrow y\}$ where $x \cup y = I$ and $|x| = 1$. The set of* max features *M_I is $ante(R_I)$.*

LEMMA 1 (CONF. WEAK DOWNWARD CLOSED). *Let M_I be the set of max features derived from I. Then any subset* *of I which contains an element of M will have a confident rule whose confidence is lower bounded by all rules in R_I.*

PROOF 1. *Let $i \in I \cap M_I$. Let r be a rule from I such that ante(r) $= i$ then the rule $i \Rightarrow I \cap$ cons(r) is a confident rule because:*

$$\frac{supp((I \cap cons(r)) \cup i)}{supp(i)} > \frac{supp(cons(r) \cup i)}{supp(i)}$$

DEFINITION 8 (SUB-RULES). *Given an itemset I, let R_I be the set of all rules $\{x \Rightarrow y\}$ where $x \cup y = I$. The set of* sub-rules *$SubR_I$ is the set of all rules generated from the itemset I_2 such that:*

$I \subset I_2$ and

For each $s \in SubR_I$:

$ante(s) \in ante(R_I)$

$conf(s) \geq conf(R)$

For example, the rule $A \Rightarrow B$ (confidence 90%) is a sub-rule of $A \Rightarrow B, C, D$ (confidence 80%).

By extension of Lemma 1, if the set of max features M of a node N is not empty, we can prune all child nodes of N whose itemsets are subsets of M, as we are guaranteed that such a child will only produce sub-rules of the rules generated by N.

3.3 Maximal Confident Rules

So far our approach has focused on identifying high confidence rules. We now present another property of confident rules which can be exploited to reduce the number of rules generated, without any information loss. If the set of confident rules can be restricted to that of *Maximal Confident Rules* (Definition 10), the number of rules can be significantly reduced. This approach can only be performed inline in a top-down algorithm as it exploits the way in which child nodes are constructed.

DEFINITION 9 (SUPER-RULES). *Given an itemset I, let R_I be the set of all rules $\{x \Rightarrow y\}$ where $x \cup y = I$. The set of* super-rules *$SupR_I$ is the set of all rules generated from the itemset I_2 such that:*

$I_2 \subset I$ and

For each $s \in SupR_I$:

$ante(s) \in ante(R_I)$

$conf(s) \leq conf(R_I)$

$SupR_{I_2} = \emptyset$

For example, the rule $A \Rightarrow B, C$ (90% confidence) is a super-rule of $A \Rightarrow B$ (100% confidence). However if the rule $A \Rightarrow B, C, D$ (80% confidence) exists then $A \Rightarrow B, C$ is not a super-rule.

DEFINITION 10 (MAXIMAL CONFIDENT RULES). *Let $\mathcal{R}$ be the set of confident rules from a dataset $\mathcal{D}$. The set MR of* maximal confident rules *is the set of confident rules whose super-rules are not confident. For example if the rule $A \Rightarrow$ B,C,D is not confident, but the rule $A \Rightarrow$ B,D is, then the second rule is a maximal confident rule.*

During MAXCONF, the first node N down a path which has a max feature set M of cardinality > 1 generates the maximal confident rules R. Let C_M be all child nodes of N with itemsets i such that $i \subset M$. Each child node $c \in C_M$ will generate a confident rule (not maximal) which is bounded below by the confidence of R (from Lemma 1). At this point MAXCONF outputs the confident rules of N (line 18), performs any child pruning (lines 19-21) and then continues in a depth first manner. If there remains any child nodes $c \in C_M$ after pruning, all items from M are not considered for rule generation from c (lines 23-24 and 11). Such rules are contained within R and thus can be ignored. Following this procedure, only Maximal Confident Rules will ever be generated.

3.4 MaxConf Example

In this subsection we provide a small example of MAXCONF applied to the dataset in Table 1. The complete row-enumeration tree without any pruning is shown in Figure 1(a). Incorporating the standard closure pruning (without support pruning) of RERII the tree in Figure 1(b) is produced.

Transaction	Items
1	A B C D E G
2	A C D E G
3	C D E F G H I
4	B C D E G
5	A C E G I
6	A D I
7	D I J
8	A B C D G

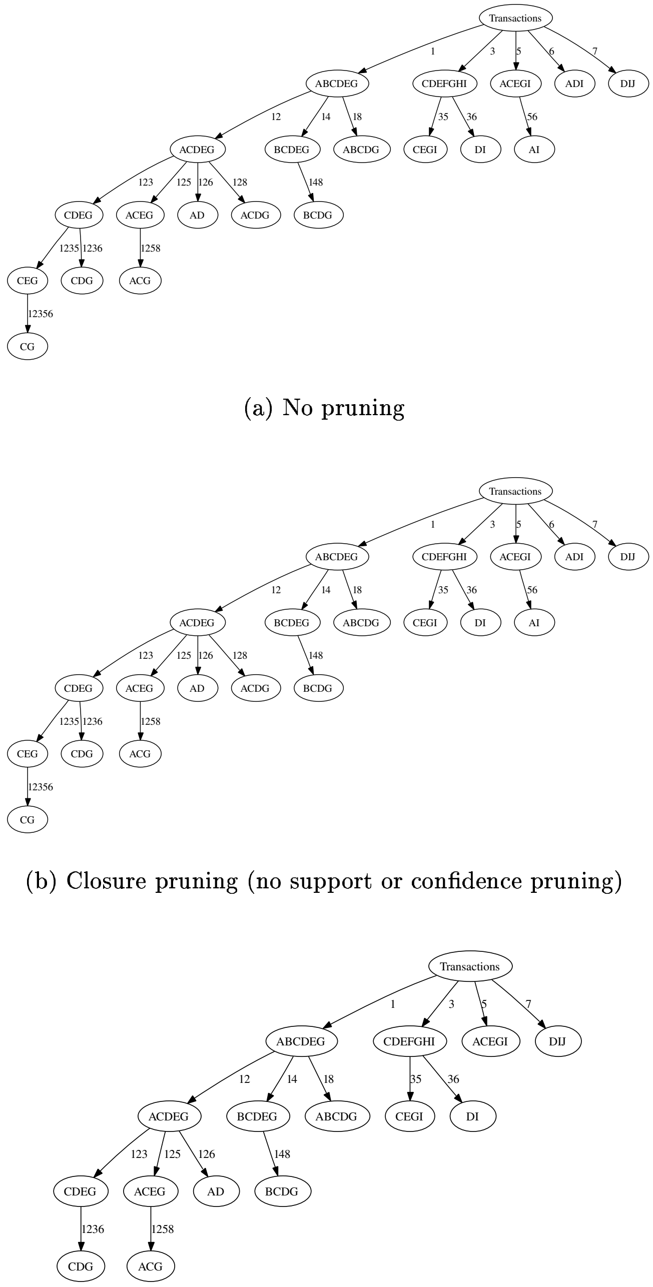
Table 1: Example transaction dataset

Suppose confidence = 2/3. Confidence Level 1 pruning will occur on Nodes AI, ADI and ACDG. For example, the support of the itemset ADI needs to be > 2.6 for this node to form any confident rules, as the minimum feature I has a support of 4.

Level 2 pruning occurs on node CEG. The parent node of CEG (CDEG) forms 3 confident rules generating the maximal feature set M = {CEG}. Therefore we know CEG will be confident as with any child nodes it may generate, and thus it can be pruned, in which case the node CG will not be created. Now suppose the support threshold is > 3. Using RERII, no rules from nodes CEGI, DI, BCDEG, DIJ, ACDEG, BCDEG or ABCDG will be generated based on support alone. Furthermore the single itemsets B, F, H and I would be immediately pruned in the first pass (unsupported). The final MAXCONF tree is shown in Figure 1(c). Table 2 shows the various rules identified by MAXCONF that can and cannot be identified using RERII with support > 3. Note due to confidence pruning, MAXCONF will not identify the rule $C \Rightarrow EG$. However this information is contained within the first rule in Table 2.

4. MAXCONF EVALUATION

In this section we concentrate on the general effectiveness of MAXCONF compared to previous methods, demonstrating the importance of pruning without support.



(a) No pruning

(b) Closure pruning (no support or confidence pruning)

(c) Confidence and closure pruning

Figure 1: MaxConf row enumeration trees

MaxConf Rule	Confidence	Support	RERII Found
C ⇒ DEG	4/6	4	Y
E ⇒ CDG	4/5	4	Y
G ⇒ CDE	4/4	4	Y
A ⇒ CG	4/5	4	Y
C ⇒ AG	4/6	4	Y
G ⇒ AC	4/6	4	Y
A ⇒ D	4/5	4	Y
B ⇒ CDEG	2/3	2	N
B ⇒ CDG	3/3	3	N
I ⇒ D	3/4	3	N
J ⇒ DI	1/1	1	N
F ⇒ CDEGHI	1/1	1	N
H ⇒ CDEFGI	1/1	1	N

Table 2: Rules identified by MaxConf and RERII

4.1 Experimental Setup

We evaluate our approach using the popular Hughes Compendium [13] of *S.cerevisiae* consisting of expression values for 6316 genes in 300 experiments (transactions) each corresponding to individual gene mutations or environment changes. For each transaction each gene's expression level is discretized into three categories: down-regulated (gene_up), up-regulated (gene_down) and neither up or down (gene_no), by binning an expression value based on the approach described in [9]. In our experiments each gene is assigned one of two individual items (gene_up and gene_down) within each transaction. Thus if in a given transaction a gene's expression level is assigned gene_no, it is not considered as an item. In total, our resulting transaction dataset consisted of 8678 different items. All experiments were performed on a PC with Pentium 4 Xeon, 1MB cache, 3.2Ghz and 4G RAM. For simplicity, all results concerning RERII are performed with support 5% unless otherwise stated. Furthermore we distinguish between MaxConf with and without the maximal restriction as simply MaxConf and MaxConf+ respectively.

4.2 Experiments

The main down fall of RERII is its inability to extract many possible association rules that meet the confidence threshold due to its support pruning. Indeed using the Hughes Compendium with 8678 items and 300 transactions, 96.5% of the single items are pruned in the first stage with a support of 10%, leaving only 301 items to be considered to form frequent itemsets and then confident rules. Without any support cut-off necessary MaxConf mines rules considering all 8678 items, and as such is capable of detecting many more rules with high confidence. Furthermore, we compared the effectiveness of mining only *maximal confident rules* (MaxConf+) to mining all high confident rules (MaxConf). As expected with a lower confidence threshold, fewer rules are generated as more maximal rules are identified. These results are summarised in Table 3.

Figure 2 provides a more detailed comparison between RERII and MaxConf, with respect to support, confidence and the maximal rule restriction. Figure 2(a) clearly highlights the drastic effects of support pruning on rule generation. When the support of RERII is lowered to zero (in an attempt to find all confident rules), no rules were ever generated as the program required too much memory. The difference in the number of rules generated by MaxConf with and without the maximal rule restriction is only moderate (Table 3), with only a 11% reduction with 85% confidence. The amount of reduction (as with any pruning) is bounded by the characteristics of the dataset. Table 4 further indicates the significant improvement of MaxConf over RERII, specifically it's capability to identify more high confident rules with a much wider support range.

Figure 2(b) shows the scalability of RERII and Max-

Conf.(%)	# Rules		
	RERII	MaxConf	MaxConf+
80	8083	21448	19090
85	3161	13181	12424
90	927	8445	8296
95	277	7229	7214
100	65	7067	7067

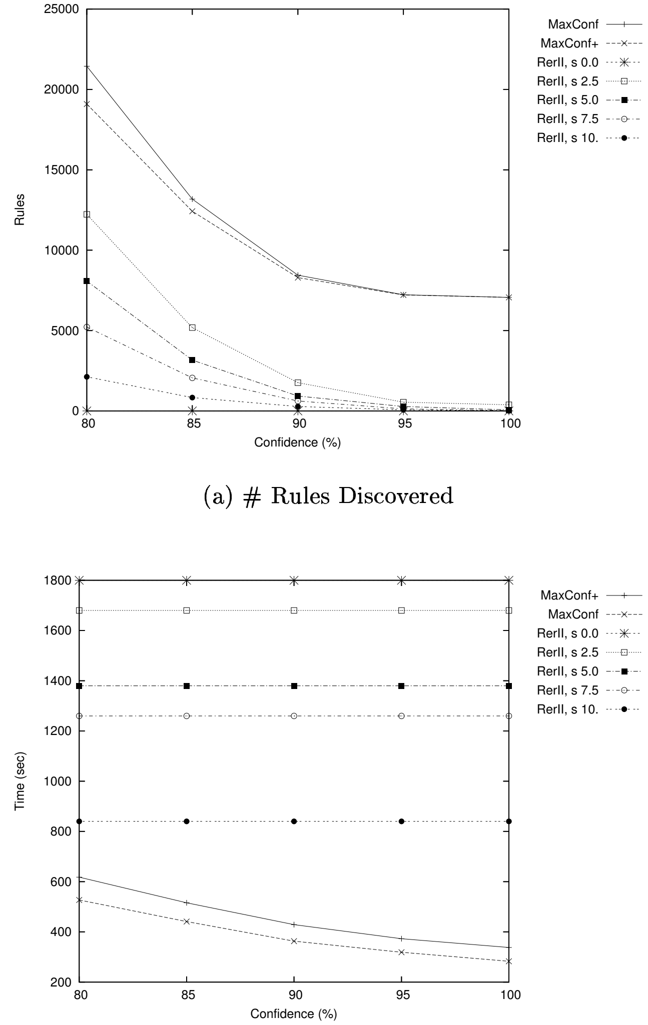
Table 3: Effect of confidence pruning

Conf.(%)	Supp. Range		
	RERII	MaxConf	MaxConf+
80	5 - 30.4	0.3 - 30.4	0.3 - 30.4
85	5 - 30	0.3 - 30	0.3 - 30
90	5 - 25	0.3 - 25	0.3 - 25
95	5 - 25	0.3 - 25	0.3 - 25
100	5 - 17	0.3 - 17	0.3 - 17

Table 4: Range of rule supports

Conf. Intuitively, with respect to support pruning, the higher the support is set, more pruning is possible and thus the run time is decreased.

Surprisingly there was a significant improvement in run time with MaxConf both with and without the maximal restriction over RERII. This was unexpected as previous approaches to pruning with confidence such as the maxPI algorithm are often less efficient then their support based alternative. The improvement is likely due to the nature of Level 2 pruning. In RERII, to generate rules satisfying confidence the complete path from the top nodes to the bottom needs to be constructed, regardless of whether a node's itemset is supported or not. However in MaxConf, when a node satisfies Level 2 pruning, all child nodes are pruned and thus it is impossible for the tree to extend further. This is indeed significantly advantageous in this case. As expected imposing the maximal restriction on MaxConf slightly increases run time, due to the extra checking required, however this approach is still more efficient than RERII.



(a) # Rules Discovered

(b) Scalability

Figure 2: Scalability and number of rules discovered with RERII, MaxConf and MaxConf+

5. RULE ANALYSIS FRAMEWORK

In this section we focus on the biological relevance of the rules we identify. We detail how biological databases can be used to validate microarray analysis algorithms, especially those designed to help generate gene networks. Firstly we concentrate on how effective our approach is in detecting known direct biological interactions in BIND. Secondly we show that many of our rules contain other gene relationships with respect to the Gene Ontology using GOstat. Finally as an example we address the *iron uptake pathway* in *S.cerevisiae*. We present some sample rules identified by MaxConf that correctly describe gene relationships in this system, which indicates the appropriateness of MaxConf to help predict gene networks.

5.1 Analysis with BIND

The Biomolecular Interaction Network Database (BIND) [5] is an online database that archives pairwise information about direct[1] interactions which can occur between two biological entities, including RNA, proteins and genes. All interactions documented are determined using traditional wet-lab experiments. Thus BIND is especially useful for analysing gene networks predicted from microarray data. Here we provide a brief overview of how our system extracts relevant data from BIND and utilises it to evaluate our rules. Firstly we determine the percentage of rules we generate that exhibit direct interactions between at least two of their items i.e. *precision* (Definition 11). The intuition behind this analysis is based on the observation that it is highly probable that for a direct interaction between two or more gene products (proteins) to occur, the expression of the genes are correlated, and hence will be present together

[1] a direct interaction refers to two biological entities physically binding together to allow some function

#	Association Rule	Supp (%)	Conf (%)
1	EUG1 ⇒ BNA2, GSC2, PDH1, TFS1, THI5, THI11, THI13, YGR043C, YML131W	1.30	100
2	SIL1 ⇒ AFR1, GSC2, YPS1, YOR289W	2.67	100
3	FRE6 ⇒ SIT1, ARN1, ARN2, ENB1, FIT2, FIT3	4.33	100
4	AKR1 ⇒ CCC2, SIT1, FTR1, ARN1, ARN2, FET3, ENB1, FIT2, FIT3	3.33	90
5	$\overline{MAC1} \Rightarrow \overline{FRE7}$	0.33	100
6	MEP2 ⇒ GLK1, GLC3, DMC1, HSP12, PRY1, NCA3, TFS1, MSC1 ,PGM2, YGP1	1	100

Table 5: Example association rules extracted using MaxConf

in at least one rule. Further we analysed the effectiveness of our approach to identify all possible interactions from the dataset, i.e. *recall* (Definition 12).

Definition 11 (Precision). *Let $\mathcal{R}$ be the set of rules identified by a mining algorithm. Let $\mathcal{B}$ be the set of pairwise direct interactions in the microarray dataset, in the form of rules. The percentage of rules which contain a direct interaction is:*

$$Precision = \frac{\#\ rules\ in\ \mathcal{R} \cap \mathcal{B}}{\#\ rules\ in\ \mathcal{R}}$$

Definition 12 (Recall). *Let $\mathcal{R}$ be the set of rules identified by a mining algorithm. Let $\mathcal{B}$ be the set of pairwise direct interactions in the microarray dataset, in the form of rules. The* recall *of direct interactions in $\mathcal{R}$ is:*

$$Recall = \frac{\#\ rules\ in\ \mathcal{R} \cap \mathcal{B}}{\#\ rules\ in\ \mathcal{B}}$$

The following steps detail how our system extracts interactions from BIND and calculates both precision and recall.

1. Extract the entire BIND dataset (BD) for the organism studied from the current exports of BIND[2]. The dataset consists of entity pairs (x, y) mapping each entity x to at least one entity y that it binds.
2. For each experiment t in the microarray, generate all possible[3] gene interactions (GI).
   (a) For each gene x in t that is up-regulated, extract all relationships from BD that x is involved in (I_x). Note that a protein interaction will only occur if its corresponding gene is expressed.
   (b) For all genes y in I_x that are also up-regulated add the pair (x, y) to GI. In total in the Hughes Compendium there are 1354 possible unique direct interactions.
3. **Precision** For each rule r :
   (a) If any pair of items in r are mapped to each other in GI than the precision count increases by 1.
4. **Recall** For each interaction (x, y) in GI:
   (a) If the items x and y appear together in at least one rule, the interaction (x, y) is said to be extracted (recalled) from the microarray experiments.

BIND analysis results are summarised in Table 6 and clearly show the effectiveness of MaxConf+ over support based mining methods. The extremely high recall (94%) is superior compared to that obtained using RERII (0.15%).

Conf. (%)	Precision (%)		Recall(%)	
	RERII	MaxConf+	RERII	MaxConf+
80	29.8	80.1	0.15	94.0
85	37.9	82.5	0.15	94.0
90	23.3	84.1	0.15	94.0
95	26.8	84.1	0.15	94.0
100	18.2	84.1	0.15	94.0

Table 6: Direct interactions identified in rules

The low recall of RERII was surprising considering the percentage of rules found that contained at least one direct interaction (37.9% with 85% confidence). After further inspection of these rules it was clear that many of the interactions were not detected as 96.5% of the genes were immediately pruned based on support during preprocessing.

Examples of rules displaying direct interactions are shown in Table 7. Both Rules 1 and 3 in Table 7 would not be identified unless the support threshold for RERII was decreased significantly (if possible). Rule 3 with 100% confidence correctly describes the relationships between the genes (CSE1 binds PCL5, which in-turn PCL5 is able to bind CRM1). Rule 2 with its high support, is the most common rule published to validate previous approaches [9]. Inspection of the rules generated by RERII showed that the majority of rules containing a direct interaction contained the items SNO1 and SNZ1.

#	Association Rule	Supp (%)	Conf (%)	BIND
1	FMP17 ⇒ ERG28 ERG25	0.60	100	ERG28: ERG25
2	CTF13 ⇒ SNO1 SNZ1	21.0	80.8	SNO1:SNZ1
3	CSE1 ⇒ CRM1 PCL5	0.33	100	PCL5:CSE1 PCL5:CRM1

Table 7: Association rules exhibiting direct interactions in *S.cerevisiae*

Many extracted association rules contain genes which interact indirectly via other genes and their products. Table 5 shows two of these rules identified using MaxConf.

In Rule 1 of Table 5 the proteins encoded by the genes GSC2, TFS1 and YGR043C all bind NUP100. THI11 binds directly to SNZ2 which binds PRP20. Each of the remaining genes bind directly to protein PRP20. Additionally, each gene in Rule 1 is involved in cellular metabolism, with the

[2] ftp://ftp.bind.ca/pub/BIND/data/datasets/taxon/

[3] a possible interaction is one that can occur under the set of experimental conditions

genes BNA2, THI5, THI11 and THI13 being specifically involved in *water soluble vitamin biosynthesis* [18] (refer to GOstat analysis).

In Rule 2 genes LSM8 and LSM2 connect genes YPS1 and SIL1 indirectly. The protein products of LSM8 and LSM2 bind directly to each other. LSM8 then directly interacts with SIL1 and LSM2 directly interacts with YPS1. Therefore, Rule 2 has successfully identified the indirect interaction between genes SIL1 and YPS1 via LSM8 and LSM2 respectively. Further analysis of Rule 2 also shows that each of the remaining proteins directly bind to the protein of STE12.

5.2 Analysis with GOstat

The Gene Ontology (GO) [18] is an international standard to annotate genes organised by their molecular function, biological process and cellular components. For every gene in the GO database there is a link to its associated gene ontologies that define its functions. The GO has a hierarchical structure starting with top level ontologies to specific descriptions with increasing depth.

GOstat [4] is a web-based query engine wrapper of the GO database where by for a group of genes as input along with various other parameters, GO annotations that are statistically over-represented within the group can be obtained. This tool provides a useful method for analysing the gene relationships we identify. To take full advantage of this query engine we developed an automated process using Python and CGI scripts to scrape the HTML results produced by GOstat for each individual itemset that generated at least one confident rule.

The number of itemsets that formed rules which contained at least two genes that were considered to be statistically over-represented by a GO are shown in Table 8. As expected, MaxConf was able to identify many more relationships.

An example of the information scraped from GOstat for Rule 6 (Table 5) is shown in Table 9. This individual rule is separated into four gene groups consisting of genes whose functions are biologically related. For instance, the genes DMC1 and MSC1 both belong to the same ontology class *meotic recombination* which has a depth of 9 within the entire GO.

Conf. (%)	#Itemsets with GO		% Itemsets with GO	
	RERII	MaxConf+	RERII	MaxConf+
80	323	899	63.5	74.9
85	187	773	65.6	78.3
90	69	693	59.4	82.7
95	34	690	60.7	82.9
100	7	690	63.6	82.9

Table 8: GO clusters identified in itemsets

5.3 Iron Uptake Pathway

In this section we further accentuate the usefulness of MaxConf for extracting correct gene relationships that depict gene networks (pathways). *S.cerevisiae* has two different mechanisms to take up iron from the external environment for it to use in other processes, which combined form the *iron uptake pathway* [10, 11]. Rules 3 - 5 in Table 5 correspond to a small sample of the rules identified by our system applicable to this pathway.

One system of the iron uptake pathway depends on a group of proteins, specifically a family of high-affinity transporters encoded by the genes ARN1, ARN2, SIT1 and ENB1. Therefore for this uptake sub-system to function each of those genes need to be co-expressed. Another sub-system of iron uptake requires some if not all the proteins FRE1-6, FET3, FIT2-3 and FTR1 [11]. MaxConf was able to detect such biological significant patterns two of which are shown in Table 5 (Rules 3 and 4). These two rules would not have been detected using the bottom-up Apriori approach to frequent itemset mining as they would have needed to be pruned with respect to support to reduce the search space.

Rule 5 in Table 5 is one of the many relationships which indicates the applicability of our approach to perturbation microarray experiments. The gene MAC1 was one of the genes chosen to be mutated in the Hughes Compendium [13]. While extracting many other gene relationships, MaxConf was able to detect relationships that Boolean networks attempt to identify. Indeed Rule 3 correctly describes the relationship between the genes MAC1 and FRE7, that is MAC1 is required to activate FRE7. Therefore FRE7 will only ever be present if MAC1 is prior [14].

6. CONCLUSIONS

A top-down algorithm MaxConf to directly discover high confidence rules was proposed. This algorithm lifts the limitation afflicting all support-based pruning methods which are unable to explore the space of low-support high confidence rules. A head to head comparison with RERII shows that MaxConf can efficiently discover more high confidence and potentially interesting rules. We provide a means to utilise biological databases for validating the relationships extracted by microarray analysis algorithms. Using this technique we highlight the importance of mining without support pruning. For example the recall of MaxConf on the biological database BIND is very high - 94% compared to 0.15% of RERII. Potential directions for future work include an analysis of the *False Discovery Rate* of MaxConf and the construction of gene networks from the discovered rules.

7. REFERENCES

[1] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 207–216, 1993.

[2] T. Akutsu, S. Kuhara, O. Maruyama, and S. Miyano. Identification of genetic networks by strategic gene disruptions and gene overexpressions under a boolean model. *Theoretical Computer Science*, 298:235–251, 2003.

[3] T. Akutsu, S. Miyano, and S. Kuhara. Identification of genetic networks from a small number of gene expression patterns under the boolean network model. In *Pacific Symposium on Biocomputing*, pages 17–28, 1999.

[4] T. Beissbarth and T. Speed. Gostat: Find statistically overrepresented gene ontologies within gene groups. *Bioinformatics*, 20(9):1464–1465, 2004.

[5] C Alfarano et.al. The Biomolecular Interaction Network Database and related tools 2005 update.

Gene Group	GO Info		
	Related GO	Depth	P-value
DMC1 MSC1	meotic recombination	9	0.0475
GLK1 GLC3 PGM2	carbohydrate metabolism	5	0.0475
HSP12 YGP1	cell communication	3	0.141
HSP12 MEP2	plasma membrane	4	0.172

Table 9: GO Information for Rule 6 in Table 5

Nucleic Acids Res, 33:D418–24, 2005.

[6] G. Cong, K.-L. Tan, A. K. H. Tung, and F. Pan. Mining frequent closed patterns in microarray data. In *4th IEEE International Conference on Data Mining (ICDM 2004)*, pages 363–366, 2004.

[7] G. Cong, K.-L. Tan, A. K. H. Tung, and X. Xu. Mining top-k covering rule groups for gene expression data. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 670–681, 2005.

[8] G. Cong, A. K. H. Tung, X. Xu, F. Pan, and J. Yang. Farmer: Finding interesting rule groups in microarray datasets. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 143–154, 2004.

[9] C. Creighton and S. Hanash. Mining gene expression databases for association rules. *Bioinformatics*, 19(1):79–86, 2003.

[10] R. F. Hassett, A. M. Romeo, and D. J. Kosman. Regulation of high affinity iron uptake in the yeast saccharomyces cerevisiae. *J Biol Chem*, 273(13):7628–7636, 1998.

[11] V. Haurie, H. Boucherie, and F. Sagliocco. The snf1 protein kinase controls the induction of genes of the iron uptake pathway at the diauxic shift in saccharomyces cerevisiae. *J Biol Chem*, 278(46):45391–6, 2003.

[12] Y. Huang, H. Xiong, S. Shekhar, and J. Pei. Mining confident colocation rules without a support threshold. In *Proc. ACM Symposium on Applied Computing (SAC)*, pages 497–501, 2003.

[13] T. Hughes et al. Functional discovery via a compendium of expression profiles. *Cell*, 102:109–126, 2000.

[14] L. J. Martins, L. T. Jensen, J. R. Simon, G. L. Keller, and D. R. Winge. Metalloregulation of fre1 and fre2 homologs in saccharomyces cerevisiae. *J Biol Chem*, 273(37):23716–23721, 1998.

[15] F. Pan, G. Cong, A. K. H. Tung, J. Yang, and M. J. Zaki. Carpenter: finding closed patterns in long biological datasets. In *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 637–642, 2003.

[16] J. Pei, J. Han, and R. Mao. Closet: An efficient algorithm for mining frequent closed itemsets. In *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pages 21–30, 2000.

[17] F. Rioult, J.-F. Boulicaut, B. Cremileux, and J. Besson. Using transposition for pattern discovery from microarray data. In *ACM SIGMOD Workshop Data Mining and Knowledge Discovery*, 2003.

[18] The Gene Ontology Consortium. The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Res*, 32:D258–D261, 2004.

[19] M. J. Zaki and C.-J. Hsiao. Charm: An efficient algorithm for closed itemset mining. In *Proc. 2nd SIAM International Conference on Data Mining*, 2002.

A Datamining Approach to Cell Population Deconvolution from Gene Expressions using Particle Filters

Sushmita Roy
Department of Computer Science
University of New Mexico
Albuquerque, NM 87131, USA
sroy@cs.unm.edu

Terran Lane
Department of Computer Science
University of New Mexico
Albuquerque, NM 87131, USA
terran@cs.unm.edu

Margaret Werner Washburne*
Department of Biology
University of New Mexico
Albuquerque, NM 87131, USA
maggieww@unm.edu

ABSTRACT

Microarrays generally measure gene expressions from a mixture of cell subpopulations in different stages of a biological process. However, little or no information about these subpopulations is actually incorporated in existing data analyses. Estimation of these subpopulation proportions is important for measuring the extent of synchrony in the entire population. Based upon the gene expression specific to individual subpopulations, genes can be clustered and assigned functions. The relative abundance of the cellular subpopulations also reveals phenotypic information of mutant populations that is valuable for studies of genetic diseases such as cancer. Thus, the quantification of subpopulation proportions is important, not only as a reliability measure of microarray data but also because of its potential relevance to functional analysis and biomedical and clinical applications.

In this paper, we describe a novel approach to model a biological process that provides (i) a maximum a posteriori (MAP) estimate of the subpopulations given the gene expression, (ii) stage-specific gene expression values and (iii) a gene clustering method based on their stage-specific expression. We have applied our approach to model the yeast cell-cycle and have extracted profiles of the population dynamics for different stages of the cell-cycle. Evaluation of statistical validity of our results using bootstrapped confidence tests reveals that our model captures significant temporal dynamics of the data. Our results are in agreement with existing biological knowledge and are reproducible in multiple runs of our algorithm.

General Terms

Population deconvolution, Cell subpopulations, Biological process, Pure stages, Gene expression.

Keywords

Particle filters, Dirichlet, Gaussian.

1. INTRODUCTION

Typically, gene-expression analysis of linear or cyclic biological processes is carried out using what are assumed to be synchronized populations of cells. Synchronized populations are obtained by arresting cells in a particular stage of the cellular life-cycle using a variety of techniques including chemical arrest, conditional mutation and nutrient starvation [8]. Synchronization is critical to the examination of pure stages that occur sequentially during the process. Unfortunately, perfect synchronization of cell populations is usually either technically difficult or impossible. As a result, most microarray analyses detect gene expression from mixtures of cells in pure stages, present as subpopulations.

Because genomic analysis of major developmental processes holds such promise, the ability to identify and analyze pure stages is crucial. The pure stage of a process gives insight into physiological changes that may affect disease prognosis and treatment. Determination of the subpopulation proportion of each pure stage provides information about the degree of synchrony, which provides insight into the arrest, recovery and initiation of a process.

Development of mathematical models for cell population deconvolution (CPD) from microarray data is very challenging since microarray data is extemely noisy and sparse. Furthermore, since very little is known about the true characteristics of pure stages and the gene expressions in these stages, these models have to be learnt in an unsupervised manner that makes minimal assumptions. This complicates the learning algorithms for these models, making them possibly intractable.

In this paper, we present an unsupervised, data-driven, probabilistic approach for mining timeseries microarray data that provides new insights about pure stages. Our approach models the biological process as a hidden-state space system, using a combination of particle filters [1, 12, 7] and expectation maximization [6], and estimates proportions of subpopulations in pure stages of a process, and the stage-specific gene expression. Using the yeast cell-cycle microarray data, we demonstrate that our approach recovers high quality estimates of the temporal evolution of the pure cell-

*Additional authors mentioned before the reference section.

BIOKDD '05, August 2005, Chicago, Illinois, USA
Copyright 2005 ACM 1-59593-213-5 ...$5.00.

cycle stages and the stage-specific gene expressions, which agree well with the existing biological knowledge of the cell-cycle stages and specific genes expressed in these stages.

2. RELATED WORK

One approach to CPD, developed by Lu *et. al*, [13], considered the observed gene expression at any point in time to be a linear combination of gene expressions from pure stages of the cell-cycle. This approach assumed that the gene expressions for a pure cell-cycle stage corresponds to a specific timepoint from an existing microarray experiment, [23], at which a required gene for that stage, had the highest expression. Therefore, this approach assumed that perfect population synchrony was maintained at those timepoints even though there were only two or three timepoints for the cell-cycle phases that usually last for 20 minutes or more.

This approach ignores the fact that cell-cycle stages extend over several minutes and therefore cannot be captured by a single timepoint. This approach may also be limited in practice due to its dependence on perfect synchronization and knowledge of such few genes. A purely data-driven approach that does not assume perfect synchronization and that provides a probabilistic estimate of the pure gene expressions is likely to be more biologically accurate and adaptable to other timeseries datasets.

Another approach developed by Bar-Joseph *et. al*, [2], is relevant to population deconvolution. This approach computes the rate of loss of synchrony and uses it to deconvolve the observed gene expression profile into pure gene expression profiles. This allows a better identification of cycling genes. However, this approach is directed towards finding cycling genes and is limited to data from cyclic biological processes such as the cell-cycle. Our approach is more general since it does not require the microarray data to be from a cyclic process.

Non-computational methods have also been used to provide measures of population synchronicity. *Fluorescent activated cell sorting* (FACS) allows qualitative analysis of population synchrony by measuring the DNA content of the cells in the population [19]. Another method uses image analysis of budding yeast images [18]. These methods can be time consuming and do not provide any information about gene expression. However, these methods can be used in concert with computational approaches to provide evidence to support predictions from mathematical models of gene expression data.

Overall, our approach has the following advantages: (a) model formulation that exploits temporal dependencies of the timeseries microarray data, (b) estimation of subpopulation proportions that provides insight into the temporal dynamics of evolving populations, (c) estimation of probability distributions of gene expression in pure stages, (d) the ability to cluster genes on the basis of stage-specific gene expression and (e) the applicability to timeseries data from both cyclic and non-cyclic biological processes, thus enabling the extraction of a wide variety of interesting gene expression dynamics including cyclic behaviour.

3. A PROBABILISTIC MODEL FOR CPD

3.1 Rationale for our model

The biological process to be modeled was considered as an ordered series of *stages*. In this paper, we are concerned with modeling the cell-cycle [20], although our method should generalize to other processes, such as sporulation, [5] and exit from quiescence, [16]. The cell-cycle comprises four phases: GapI (*G1*), Synthesis (*S*), GapII (*G2*) and Mitosis (*M*). A stage for the cell-cycle was assumed to be either a phase or a transition between phases. Based upon prior studies by Breeden *et. al*, a possible set of stages for the cell-cycle is, {*G1*, *S*, *G2/M*, *M*, *M/G1*}, where *G2/M* implies transition between *G2* and *M*, and *M/G1* implies transition between *M* and *G1*.

We assume that each subpopulation is in one of the stages of the biological process. Hence, stage and subpopulation are analogous to each other. We assume that cells transition from one stage to another causing a change in the subpopulation proportions as a function of time. These changing proportions, in turn, affect the overall gene expression pattern. The observed gene expression at every timepoint, then, is a function of (a) the pure or stage-specific gene expression from cells in each subpopulation and (b) the proportion of that subpopulation.

To model the biological process, we need to specify the mechanism by which the proportions change with time, i.e., the *process model*, and the function which determines the observed gene expression from the proportions, i.e., the *observation model.* After fixing the forms of the process and observation models, we use *unsupervised learning* based on the combination of particle filters and expectation maximization, to probabilistically estimate the subpopulation proportions and the stage-specific gene expressions, based solely on the observed timeseries data. Thus, we solve the CPD problem with an unsupervised approach that makes minimal assumptions and does not heavily rely on prior knowledge.

3.2 Model overview

Our approach models the biological process as a dynamic hidden state-space system, that estimates the posterior probability distribution of the hidden state given the observed expression data. Similar to standard dynamic state-space systems [22], our approach specifies (a) a process model for modeling the state transition between successive timepoints, (b) an observation model for modeling the relationship between state and observation (gene expression) at each timepoint, and (c) a prior probability distribution for the hidden state at each timepoint. Thus, given the prior and the observation at every timepoint, the solution to the CPD problem is the posterior distribtion of the hidden state.

Unlike in well understood systems such as hidden Markov models (HMMs) [21] and Kalman filters [9], the hidden state in our approach is constrained to be a *multinomial vector* that specifies the percentages or proportions of subpopulations in pure stages of a process. A natural choice for a prior distribution for multinomials is a Dirichlet [17]. From our recent work we have found that the analytical form of the resulting posterior is a mixture of Dirichlet distributions, wherein the number of components increases exponentially. This crucial point that the hidden state of our model is a multinomial rather than a discrete scalar (HMMs) or an unconstrained vector (Kalman filters), results in difficult parameter estimation and inference problems.

To render these problems tractable we (a) assume a Dirichlet prior distribution over the hidden state at each timepoint [17]; (b) employ a *particle filter* based algorithm for inferring the hidden state posterior distribution [1, 12, 7]; (c) assume that individual gene expressions are Gaussianly distributed and independent given the hidden state; and (d) use the Expectation Maximization (EM) algorithm for parameter estimation [6].

3.3 A generative model of microarray data

The microarray time series data in our analysis was first transformed to intensity ratios from the logarithm of the intensity ratios. Hereafter we refer to *gene expression* as the intensity ratio for a single gene and an *expression vector* as the ratios for all the genes at a specific timepoint. We use the terms, stage, biological stage and pure stage to imply the same thing: a stage of a biological process.

Let x_t denote the hidden state vector at time t, for $1 \leq t \leq T$, where T denotes the total number of timepoints in the microarray timeseries. The n components of x_t, denoted by $x_t(k)$ for $1 \leq k \leq n$, quantify the proportions of the subpopulations in the n different stages of the biological process at time t. For example, for the cell-cycle, $x_t(k)$ corresponds to a subpopulation in one of the stages, *G1, S, G2/M, M* or *M/G1*. Even though n, or the number of stages, is known prior to our training process, we do not know which $x_t(k)$ corresponds to which stage, i.e. our results are unique only up to label (name of a biological stage) permutations. The mechanism of determining the actual biological stage names is described in Section 5.3. Since x_t takes the form of a multinomial vector, it obeys the constraints, $0 \leq x_t(k) \leq 1; \forall k$, and $\sum_k x_t(k) = 1$.

Let y_t denote the expression vector. The m components of y_t are the expression ratios of the m individual genes, denoted $y_t(j)$ for $j = 1, \ldots, m$. The CPD problem, then, is: given a time series of expression vectors, $\{y_1, \ldots, y_T\}$, estimate the hidden state describing the subpopulation distribution at every timepoint, $\{x_1, \ldots, x_T\}$.

We assumed that the hidden state, x_t, is conditionally dependent only on the preceeding state, x_{t-1} and that the observed gene expression y_t, is conditionally dependent only on the corresponding hidden state, x_t. This yields a temporal process similar in statistical structure to a hidden Markov model, with a first-order Markov process model, $P(x_t|x_{t-1})$ and an observation model, $P(y_t|x_t)$.

The observation model (Section 3.3.3) specifies the stage-specific gene expression for every gene, j in a pure stage, k by a Gaussian, $N(\mu_k^j, \sigma_k^j)$, where μ_k^j and σ_k^j are the mean and standard deviation for the expression of j^{th} gene in the k^{th} stage. The process model (Section 3.3.2), is specified by a transition matrix, A. Thus our complete model is specified by the parameters: stage dependent Gaussians for the gene expression and the transition matrix.

The learning algorithm for the model parameters is based on Expectation Maximization (EM) (Section 5). The algorithm uses a "forward step" to estimate $P(x_t|y_1, \ldots, y_t)$ and a "backward step" to estimate $P(x_t|y_{t+1}, \ldots, y_T)$. These two estimates are combined to derive the transition probabilities in A. The Gaussian parameters for the stage dependent gene expressions are assigned their maximum likelihood estimates. However, the algorithm needs to account for the complex form of the hidden state distribution that arises because the hidden state is a multinomial vector. Hence, we use particle filters (Section 4), which allows us to approximate the probability distributions in the forward and backward step by a set of weighted samples (Section 5.1).

After running the EM procedure to convergence, we have a complete model of the biological process. The solution to our original deconvolution problem: the proportions of the individual subpopulations for every timepoint t, is then specified by the expected value of hidden state posterior distribution, $P(x_t|y_1, \ldots, y_t)$. Different components of our model are detailed in the following sections.

3.3.1 Modeling the hidden state posterior:

The posterior distribution of hidden state, $P(x_t|y1, \cdots, y_t)$ is approximated by a set of samples, $\{x_t^1, \ldots, x_t^N\}$, and the set of sample weights, $\{w_t^1, \ldots, w_t^N\}$, where N denotes the total number of samples and is specified by the user. These samples are generated from a probability distribution of a known parameterized form.

3.3.2 Process model:

The process model specifies $P(x_{t+1}|x_t)$, the conditional probability distribution of the next state, x_{t+1}, given the current state, x_t. We assume that the mechanism by which x_t transforms to x_{t+1} is dependent on the probability with which a cell in the k^{th} pure stage at time, t, transitions to the l^{th} pure stage at time, $t+1$. This stage transition probability is specified by the transition matrix, A. Hence, the element, $A(l, k)$ specifies the probability with which a cell in the k^{th} stage makes a transition to the l^{th} stage at the next timepoint, where l and k denote a row and a column respectively.

Given the sample set, $\{x_t^1, \ldots, x_t^N\}$ that approximates the probability distribution over x_t, we use the process model to obtain the sample set, $\{x_{t+1}^1, \ldots, x_{t+1}^N\}$ that approximates the probability distribution over x_{t+1}. This is done by projecting every sample, x_t^i, to $x_{t+1}^i = Ax_t^i$.

3.3.3 Observation model:

The observation model specifies $P(y_t|x_t)$, the conditional probability distribution of the current observation, y_t, given the current state, x_t. The expression of the j^{th} gene, in the k^{th} pure stage, is assumed to be modeled by a Gaussian, $N(\mu_k^j, \sigma_k^j)$. Since the actual population from which mRNA is extracted, is a mixture of subpopulations, the observed gene expression is modeled by a mixture of Gaussians. Hence, $P(y_t(j)|x_t) = \sum_{k=1}^n x_t(k) P(y_t(j)|\mu_k^j, \sigma_k^j)$, where $y_t(j)$ is the observed expression ratio of the j^{th} gene at time t and $x_t(k)$ is the contribution of the k^{th} Gaussian for the subpopulation in the k^{th} stage.

We further assumed a naive Bayes model for the gene expressions – expression of a gene is dependent only on the stage of the cell. Given the stage, the gene expressions are conditionally independent of each other. The conditional probability, $P(y_t|x_t)$, is then a product of probabilities of individual gene expressions, i.e., $P(y_t|x_t) = \prod_{j=1}^m P(y_t(j)|x_t)$. These assumptions may not be ideal for modeling gene expressions, but our current focus is to identify cell populations in different stages and modeling gene relations within a stage is unlikely to effect our deconvolution results. However, we are considering better models such as tree-augmented networks for future work.

4. PARTICLE FILTERS

Particle filters are used to approximate probability distributions that have an unknown or complex analytical form, by a set of weighted samples, and can be used for the hidden state inference in non-linear dynamic systems [1]. Samples are drawn from a parameterized probability density called the *importance density.* The weight of each sample is proportional to the likelihood of the observed data given that sample. In the following subsections we describe the particle filter used in our approach, the importance density selection and the sample weight calculation.

4.1 Sample Importance Resampling filter

A well known particle filter for hidden state-space systems is *Sample Importance Resampling* (SIR) filter [10], wherein the prior, $P(x_{t+1}|x_t)$, is chosen as the importance density. The SIR filter begins with a set of samples, $\{x_t^1, \ldots, x_t^N\}$, drawn from the importance density, followed by the calculation of the sample weights, $\{w_t^1, \ldots, w_t^N\}$. The sample weight, w_t^i is determined by the likelihood, $P(y_t|x_t^i)$, of the observation, y_t, given the i^{th} sample, x_t^i. The posterior distribution of the hidden state given the observation is represented by a subset of these samples, chosen by a *re-sampling* step. The probability that a sample is selected by the re-sampling step is proportional to its weight. The chosen samples are then projected using a *projection* step, determined by the process dynamics.

4.1.1 Sample degeneration

One of the problems that we encountered with the SIR filter was rapid sample degeneration. This was because a small subset of the entire sample set had very high weights compared to the rest. The re-sampling step repeatedly selected these samples, resulting in the degeneration of all or most of the samples to single points. This is a known issue with the SIR filter and there are several smoothing strategies to alleviate this problem [4, 24].

We employed the *jittering* technique to handle sample impoverishment. We added a small amount of perturbation to every sample, x_t^i, selected by the re-sampling step. This perturbation was a sample drawn from a Gaussian, $N(0, \omega)$. Thus if x_t^i were selected p times, it would be represented by a Gaussian, $N(x_t^i, \omega)$, estimated from the p different samples. The value of ω controls the extent of jitter and is chosen experimentally so as to solve the degeneracy problem and yet prevent the system from diverging from its true behaviour. We found $0.05 \leq \omega \leq 0.1$ to be a reasonable range of values for the jitter.

4.2 Importance density

Since x_t is a multinomial vector, a natural choice of the importance density is a Dirichlet. Choosing a Dirichlet has the added advantage of allowing the incorporation of biological knowledge of the cell population. For example, the Dirichlet parameters for the prior state distribution at time, $t = 1$, can be assigned on the basis of existing knowledge that populations are synchronized to 80% or more.

4.3 Sample weight calculation

The sample weight, w_t^i, of the sample, x_t^i, is $P(y_t|x_t^i)$ and is given by the observation model as follows:

$$w_t^i = \prod_{j=1}^{m} \sum_{k=1}^{n} x_t^i(k) P(y_t(j)|\mu_k^j, \sigma_k^j) \qquad (1)$$

This is similar to $P(y_t|x_t)$ in Section 3.3.3, with x_t being replaced by x_t^i. The rationale for replacing x_t by x_t^i is that while x_t represents the exact hidden state, x_t^i represents a possible description of it: x_t^i is a sample from the posterior distribution that measures the uncertainty about x_t^i.

Algorithm 1 Forward particle filtering algorithm

```
1. For $t = 1$, draw samples from the Dirichlet prior that is initialized either randomly or on the basis of known information of cell populations.
2. Calculate weights of samples at $t$ using eqn. (1).
3. Execute the re-sampling step to select samples based on the sample weights. These samples estimate $P(x_t|y_1, \ldots, y_t)$.
4. Execute the projection step using the process model to obtain samples for the next timestep, $t + 1$.
5. Increment $t$ by 1.
6. Repeat steps 2-5 for all $t \leq T$
```

5. MODEL TRAINING AND PARAMETER ESTIMATION

As described in the preceeding sections, the model parameters are the transition matrix, A, for the process model and the Gaussian parameters, $N(\mu_j^j, \sigma_j^k)$, which specify the observation model. Parameter estimation is done using a learning procedure based on Expectation Maximization (EM), comprising an Expectation (E) step and a Maximization (M) step.

5.1 Expectation (*E*) step

The hidden variables in the system are the state variables, x_t. During the E step we use the model parameters and obtain the expected values for x_t. The E step is similar to the expectation step of the Baum-Welch (BW) algorithm [21], and comprises a *forward* step and a *backward* step. Unlike the HMM that uses discrete, univariate random variables for state and observation, our learning problem is complicated by the fact that x_t and y_t are multivariate continuous random variables and that x_t is a multinomial. To handle this problem, all probability densities in both the forward and backward steps are represented by the weighted samples from the particle filter.

5.1.1 Forward step:

The forward step (Algorithm 1) estimates the conditional probability of a state, x_t, given the observations, $\{y_1, \ldots, y_t\}$, i.e., $P(x_t|y_1, \ldots, y_t)$. This differs from the forward step of BW, which calculates the joint probability of a state at time, t and the set of observations upto time t.

5.1.2 Backward step:

The backward step (Algorithm 2) estimates the conditional probability of a state, x_t, given the observations $\{y_{t+1}, \ldots, y_T\}$, i.e., $P(x_t|y_{t+1}, \ldots, y_T)$. Again, our backward step is different from BW since the latter calculates the conditional probability of the partial observation sequence, $\{y_{t+1}, \ldots, y_T\}$ given a state at time t. Recall that the k^{th} component of x_t, $x_t(k)$, is the probability of being in the k^{th} pure stage of the biological process.

Let $\bar{x}_t^i$ represent the samples which approximate $P(x_t|y_{t+1}, \ldots, y_T)$. Given $\bar{x}_{t+1}^i$, the i^{th} backward sample from time, $t + 1$, $\bar{x}_t^i$ is recursively obtained in two steps. In the first step, each

$\bar{x}_t^i(k)$ is calculated using eqn. (2),

$$\bar{x}_t^i(k) = \sum_{l=1}^{n} A(l,k) e_l(t+1) \bar{x}_{t+1}^i(l) \qquad (2)$$

where e_l is a multinomial distribution describing the emission probability for the T expression vectors for the l^{th} stage. Every element, $e_l(t)$ is calculated using:

$$e_l(t) = \frac{1}{Z} \prod_{j=1}^{m} P(y_t(j) | \mu_l^j, \sigma_l^j) \qquad (3)$$

where Z is a normalization term. In the second step, each $\bar{x}_t^i(k)$ is normalized by $\sum_{k=1}^{n} \bar{x}_t^i(k)$ to result in a multinomial vector.

Algorithm 2 Backward particle filtering algorithm

```
1. Generate samples from a uniform Dirichlet (all parameters set to 1) for time, $t = T$.
2. Calculate weights for the samples using eqn. (1); resample according to weights.
3. Decrement $t$ by 1.
4. Obtain samples for $t$, $\bar{x}_t^i$, from the selected samples, $\bar{x}_{t+1}^i$, at $t + 1$ using eqn. (2). These samples estimate $P(x_t | y_{t+1}, \ldots, y_T)$.
5. Repeat steps 2-4 for all $t$.
```

5.2 Maximization (*M*) step

The M step uses the expected values of x_t to estimate the model parameters A, μ_k^j and σ_k^j, $1 \leq k \leq n$ and $1 \leq j \leq m$.

5.2.1 Estimation of A:

Every element, $A(l,k)$, specifies the probability with which a cell in the k^{th} stage at time, t, makes a transition to the l^{th} stage at time $t+1$. Our technique for estimating A is a slight modification of the standard HMM.

Estimation of A requires two variables, $\xi_{kl}^i(t)$ and $\gamma_k^i(t)$. $\xi_{kl}^i(t)$ specifies the probability of a cell being in the k^{th} stage at time, t, *and* the l^{th} stage at time, $t + 1$. $\gamma_k^i(t)$ specifies the probability of a cell being in the k^{th} stage at time, t. The superscript, i indicates that these variables need to be estimated for every sample. Assuming that A is initialized to some value, $\xi_{kl}^i(t)$ is calculated from the components of forward sample, x_t^i, and the backward sample, $\bar{x}_{t+1}^i$ as follows:

$$\xi_{kl}^i = \frac{1}{Z'} x_t^i(k) A(l,k) \bar{x}_{t+1}^i(l) e_l(t+1) \qquad (4)$$

where, Z' is a normalization term and $e_l(t+1)$ is the emission probability of the observation, y_t from the k^{th} stage. Given $\xi_{kl}^i(t)$, $\gamma_k^i(t)$ is then, $\sum_{l=1}^{n} \xi_{kl}^i(t)$.

Once the $\xi_{kl}^i(t)$'s and the $\gamma_k^i(t)$'s have been calculated for each of the N samples, the element, $A(l,k)$ is calculated as follows:

$$A(l,k) = \frac{\sum_{t=1}^{T} \sum_{i=1}^{N} \xi_{kl}^i(t)}{\sum_{t=1}^{T} \sum_{i=1}^{N} \gamma_k^i(t)} \qquad (5)$$

This is followed by normalizing every column $A(:,k)$ such that it is a probability vector.

5.2.2 Estimation of μ_k^j and σ_k^j:

The Gaussian parameters, μ_k^j and σ_k^j, which specify the stage-specific gene expression, are set to their maximum likelihood (ML) estimates for a mixture of Gaussians [11]. μ_k^j, the mean expression of the j^{th} gene in the k^{th} pure stage is estimated from the observed expression ratio of the j^{th} gene, $y_t(j)$, and the forward samples, x_t^i, for all timepoints, $1 \leq t \leq T$. The component $x_t^i(k)$ specifies the contribution of the k^{th} Gaussian to the observed expression, $y_t(j)$. Hence, μ_k^j is given by

$$\mu_k^j = \frac{\sum_{t=1}^{T} \sum_{i=1}^{N} x_t^i(k) y_t(j)}{\sum_{t=1}^{T} \sum_{i=1}^{N} x_t^i(k)}. \qquad (6)$$

σ_k^j is calculated using a similar modification to the standard ML formula.

5.3 Training convergence

The EM algorithm is said to have converged when the log likelihood of the observed data, $L = P(y_1, y_2, \ldots, y_T)$, changes only by some small value, ϵ. Using the chain rule, $L = P(y_1) \prod_{t=2}^{T} P(y_t | y_1, \ldots, y_{t-1})$. Then using an approximation described in [7], the first term is calculated using $P(y_1) = \sum_{i=1}^{N} P(y_1 | x_1^i) P(x_1^i)$. The rest of the product is calculated as $P(y_t | y_1, \ldots, y_{t-1}) = \sum_{i=1}^{N} P(y_t | x_t^i) \hat{w}_{t-1}^i$ where, $\hat{w}_t^i$ is the normalized sample weight for x_t^i.

Once the training algorithm converges, we have the sample set, $\{x_t^1, \ldots, x_t^N\}$, which is an approximation of posterior distribution, $P(x_t | y_1, \ldots, y_t)$; and the probability density of stage-specific gene expressions, $N(\mu_k^j, \sigma_k^j)$. The mean value of $\{x_t^1, \ldots, x_t^N\}$, $\hat{x}_t$, is then used as our estimate of the subpopulation proportions of the stages at every timestep, t, for $1 \leq t \leq T$.

The stage-specific gene expression is used for assigning a gene to one of the n stages. The j^{th} gene is assigned to the stage $k = \arg\max_l \mu_l^j$, that is the stage with the highest mean value of all the stage-specific expressions for that gene. Genes associated with the same stage (member genes) are therefore clustered together based on their stage-specific expression value.

6. RESULTS

We applied our approach to perform population deconvolution on microarray timeseries of the yeast cell-cycle (Table 1). The yeast cell-cycle is a well-studied biological process required for cell growth and cell division [20]. Since it is a cyclic biological process, it is an ordered set of stages that are repeating in the order, *G1* → *S* → *G2/M* → *M* → *M/G1*, at periodic time intervals (the cell-cycle).

For all three datasets, the number of stages, n, was set to 5, corresponding to the cell-cycle phases, *G1*, *S* and *M* and the transition between phases, *G2/M* and *M/G1*. The number of genes, m, was at most 712. Of the 712 genes, 696 genes were studied by [13]. The remaining 16 genes were added on the basis of biological literature.

Recall from Section 3.3 that each component of x_t, $x_t(k)$, specifies the subpopulation proportion for the k^{th} stage at time, t. The mapping from index, $1 \leq k \leq n$, to cell-cycle stages, $\{G1, S, G2, M, M/G1\}$, however is unknown. This is the result of the unsupervised learning method that we employ – the deconvolution results are unique only up to a label (name of a biological stage) permutation. To resolve this ambiguity, we use additional biological knowledge, [23,

Table 1: Yeast cell-cycle datasets

Dataset	Experimenter	Synchronizing method	Genes	Timepoints	Time interval
S_a	Paul Spellman	α-factor	712	18	7min
L_a	Linda Breeden	α-factor	706	13	10min
S_c	Paul Spellman	*cdc15-2*	710	24	Some at 10min and some at 20min

L_a and S_c did not have expression measurements for 6 and 2 genes respectively

Table 2: Number of known, required cell-cycle genes

Biological Stage	Number of genes
$G1$	19
S	21
$G2/M$	12
M	7
$M/G1$	19

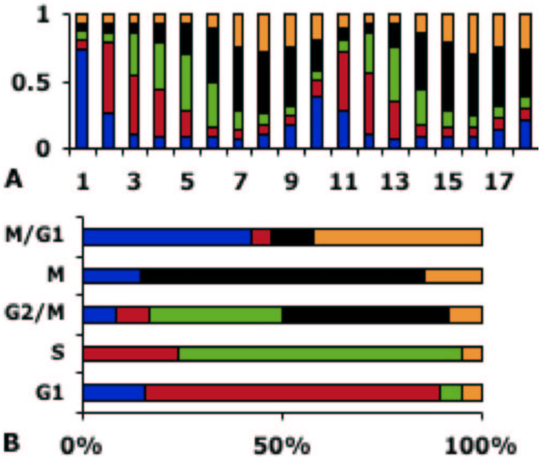


Figure 1: A. Multinomial vectors for subpopulation distributions for all 18 timepoints from S_a (α-factor experiment done by Spellman *et. al*). B. Percentage of known genes present in different member gene sets.

3, 14, 15], about genes known to be highly expressed in the cell-cycle stages (Table 2). The k^{th} stage represents a specific cell-cycle stage, if the member gene set of the k^{th} stage has the largest percentage of the known genes for that cell-cycle stage.

The deconvolution results for the datasets, S_a, L_a and S_c are presented in Figs. 1, 2 and 3, and analyzed in Sections 6.1, 6.2 and 6.3 respectively. Each figure comprises two parts, A and B. A is a column graph of the subpopulation proportions (y-axis) for the five stages (different colors) at every timepoint (x-axis). The proportion for the k^{th} subpopulation at time, t is specified by the k^{th} component, $\hat{x}_t(k)$, of the sample mean, $\hat{x}_t$. B shows the percentages of known genes for a cell-cycle stage that were present in the member genes for each of the five stages. The percentages and the cell-cycle stages are along the x-axis and the y-axis respectively. For example, Fig. 1C shows that of the 19 genes known for *G1*, 16% were present in the member genes of the blue stage, 74% were present in the member genes of the red stage and so on.

6.1 Results using S_a:

We identified subpopulations for five stages that exhibited cyclic patterns, Fig. 1A. The subpopulation proportions of individual stages peaked at certain timepoints, indicating that at that time, cells in that stage represent the largest subpopulation, e.g. blue peaked at $t = 1$ and $t = 9$. The relative proportions also illustrated that none of the timepoints had completely synchronous populations with all cells in one pure stage. We found that known genes for the phases, *G1*, *S* and *M* were predominantly present in member genes from three different stages ($\approx$ 74% (red), 71% (green) and 71% (black) respectively), Fig. 1B. Majority of the known genes for *G2/M* and *M/G1* were distributed between member genes of two stages (green and black for *G2/M* and black and orange for *M/G1*). This allowed the following mapping between the stages in our model and the cell-cycle stages: red:*G1*, green:*S*, green and black:*G2/M*, black:*M*, blue and orange:*M/G1*. The ordering of the peaks for the stages, was in agreement with the known ordering of the cell-cycle stages, further supporting the biological relevance of our results.

The inability to assign a single subpopulation for *G2/M* and *M/G1* may indicate that there is so much overlap between the corresponding phases (*G2* and *M*, *M* and *G1*), that they cannot be distinguished into separate stages of our model.

The most significant Gene Ontology (GO) processes (`http://www.yeastgenome.org/`) for member genes of the orange stage were related to α-factor response. The same stage also had a peak at timepoint, $t = 1$. This implied that the major contributor of the observed gene expression at $t = 1$ is the subpopulation of cells responding to α-factor. This is biologically significant since the synchronizing method was α-factor.

Lu *et. al*, [13] used the timepoints, $t = 3, 5, 6, 7$ and 10, from this dataset to represent pure gene expression for the cell-cycle stages, $G1, S, G2, M, M/G1$ respectively. However, our results indicate that the observed expression vector at all timepoints are derived from mixed subpopulations. Thus the assumption of these timepoints representing pure cell-cycle expression is unlikely to be accurate, especially since these stages extend over several minutes.

6.2 Results using L_a:

We identified cyclic profiles for five subpopulations, Fig. 2A. All cell-cycle stages other than *G2/M* and *M/G1* separated into different stages of our model, Fig. 2B, resulting in a similar association between the stages of our model and the cell-cycle stages as in S_a. The ordering of the cell-cycle stages was preserved in the ordering of the peaks, Also, majority of the cells at $t = 1$ were involved in α-factor response.

6.3 Results using S_c:

We identified cyclic profiles for five subpopulations in dif-

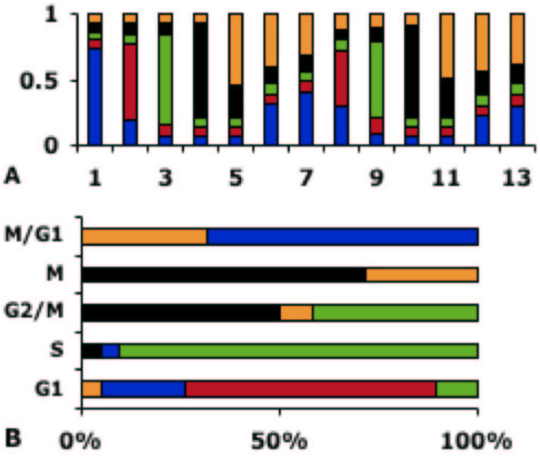


Figure 2: A. Multinomial vectors for subpopulation distributions for all 13 timepoints from L_a (α-factor experiment done by Dr. Breeden's laboratory). B. Percentage of known genes present in different member gene sets.

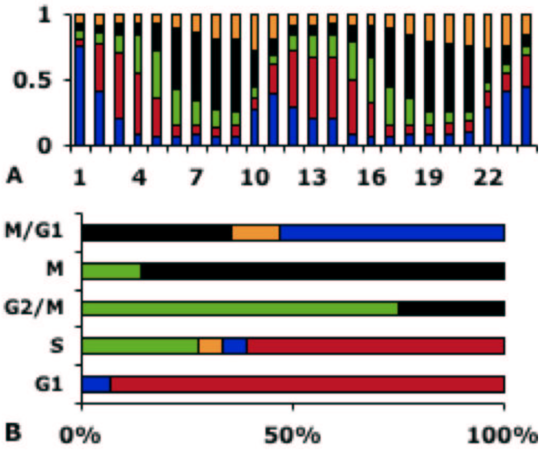


Figure 3: A. Multinomial vectors for the subpopulation distributions for all 24 timepoints from S_c (*cdc*15-2 experiment done by Spellman *et. al*). B. Percentage of known genes present in different member gene sets.

ferent biological stages, Fig. 3A. The peak at timepoint, $t = 1$, corresponded to the blue stage with member genes involved in M and $M/G1$. This makes biological sense since *cdc*15 arrests cells in late M phase. However, most of our predicted subpopulations overlapped with two cell-cycle stages, Fig. 3B. This maybe due to lesser synchrony in this dataset as compared to S_a or L_a.

6.4 Statistical validation of results

To judge the statistical significance of our results, we tested our algorithm for consistency of results and preservation of temporal dependencies. To check for consistency of the predicted subpopulation proportions, we trained r different models for every dataset z, $z \in \{S_a, L_a, S_c\}$, resulting in r predictions for each subpopulation. The standard deviations of these predictions (≤ 0.08) for a particular z indicated that they were reasonably close.

To test whether our approach preserves temporal dependencies, we used the *bootstrapped confidence test* [11]. Let Q represent a model trained on data with temporal dependencies, i.e. z, and let R represent a model trained on randomly shuffled data z_s, where z_s is obtained by reordering the observation vectors in z. To compare Q and R, we calculated the log likelihood of the observed data, z (refer to Section 5.3), for both Q and R. Let ρ_z and τ_z denote the average likelihood and standard deviation from Q for a particular dataset $z, z \in Z$. Let ρ_{z_s} and τ_{z_s} denote the average likelihood and standard deviation of z from R. These averages are obtained by starting with different initializations of the model parameters at the beginning of the EM training. We found that ρ_{z_s} was significantly smaller than ρ_z, indicating that the incorporation of temporal dependencies makes Q a better model for the observed data as compared to R. Table 3 describes the results for the confidence test.

Table 3: Likelihood mean and standard deviations

z	ρ_z	τ_z	ρ_{z_s}	τ_{z_s}
S_a	-738.7503	130.7760	-2612.045	286.0214
S_b	-6392.7	132.8753	-8371.642	332.229
L_a	2915.2406	23.2397	-1033.535	627.773

z refers to a dataset, ρ_z and τ_z are mean and stdevs for original z, ρ_{z_s} and τ_{z_s} are mean and stdevs for shuffled z.

7. CONCLUSION

In this paper we have described a particle filter based framework for extracting meaningful infomation about subpopulations from microarray data. The key ideas in our approach are (i) formulation of CPD as a dynamic hidden-state system that models the temporal dependencies in the timeseries data and (ii) constraining the hidden state to be a multinomial vector, thus allowing the direct quantification of subpopulation proportions. Our learning algorithm uses a combination of particle filters and Expectation Maximization. On convergence we have the estimate of the hidden state distributions and the pure stage-specific gene expressions.

The application of our approach to the yeast cell-cycle data demonstrated that (i) we could obtain cyclic profiles of subpopulations corresponding to cell-cycle stages, (ii) we could characterize these stages in terms of probability distributions of pure gene expression, and (iii) none of the microarray timeseries were composed of completely synchronous populations. Although the biological validation of our results is somewhat preliminary, inspite of the unsupervised nature of the decovolution problem and the limited knowledge of the true nature of pure populations, the outcome of the statistical validation tests are encouraging.

8. FUTURE WORK

We want to further investigate the analytical form of the hidden state posterior distribution and provide error bounds between its approximate and true value. We want to apply our approach to characterize subpopulations from other biological processes such as sporulation ,[5], and exit from quiescence, [16]. Preliminary work in this direction has been very encouraging. We also want to use results from *in-vivo* experiments in concert with our method to provide more biological insight into our results. One such experiment provides the *budding index count* that rises dramatically at the onset of the Synthesis phase of the cell-cycle.

All of these future goals are highly relevant to biological data mining since they will enable us to reinterpret single timepoint expression data in terms of the pure stages from different biological processes. The knowledge gained from the relative abundance of subpopulations from such datasets will provide valuable insight for analyzing aberrant gene expression patterns in human diseases.

9. ACKNOWLEDGMENTS

We thank Drs. Linda Breeden and Pramila Tata for making their dataset available to us and Drs. Peter Wentzell and Sonia Santa Anna-Arriola for helpful discussions. This work was supported by grants from the NSF (MCB-0092364) and NIH (HG02262).

10. ADDITIONAL AUTHORS

Additional authors: Chris Allen (Department of Biology, email: cpallen@unm.edu) and Anthony D. Aragon (Department of Biology, email: adaragon@unm.edu).

11. REFERENCES

[1] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 2002.

[2] Z. Bar-Joseph, F. Z., D. K. Gifford, I. Simon, and R. Rosenfeld. Deconvolving cell cycle expression data with complementary information. *Bioinformatics*, pages R586–R588, 2004.

[3] L. L. Breeden. Cyclin transcription: Timing is everything. *Current Biology*, 2000.

[4] J. Carpenter, P. Clifford, and P. Fearnhead. Building robust simulation-based filters for evolving data sets, 1999.

[5] S. Chu, J. DeRisi, M. Eisen, J. Mulholland, D. B. D, P. O. Brown, and I. Herskowitz. The transcriptional program of sporulation in budding yeast. *Science*, pages 699–705, 1998.

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 1977.

[7] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering, 1998.

[8] B. Futcher. Cell cycle synchronization. *Methods in Cell Science*, 1999.

[9] Z. Ghahramani and G. E. Hinton. Parameter estimation for linear dynamical systems. Technical Report CRG-TR-96-2, University of Toronto, 1996.

[10] N. Gordon, D. Salmond, and A. Smith. Novel approach to nonlinear /non-Gaussian bayesian state estimation. *IEE Proceedings-F (Radar and Signal Processing)*, 1993.

[11] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001.

[12] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *Internation Journal of Computer Vision*, 1998.

[13] P. Lu, A. Nakorchevskiy, and E. M. Marcotte. Expression deconvolution: A reinterpretation of DNA microarray data reveals dynamic changes in cell populations. *Proceedings of the National Academy of Sciences of the United States of America*, 2003.

[14] V. L. Mackay, B. Mai, L. Waters, and L. L. Breeden. Early cell cycle box-mediated transcription of CLN3 and SWI4 contributes to the proper timing of the G1-to-S transition in budding yeast. *Molecular and Cellular Biology*, 2001.

[15] B. Mai, S. Miles, and L. L. Breeden. Characterization of the ECB binding complex responsible for the M/G1-specific transcription of CLN3 and SWI4. *Molecular and Cellular Biology*, 2002.

[16] M. J. Martinez, S. Roy, A. B. Archuletta, P. D. Wentzell, S. S. Anna-Arriola, A. L. Rodriguez, A. D. Aragon, G. A. Quiones, C. Allen, and M. Werner-Washburne. Genomic analysis of Stationary-Phase and exit in Saccharomyces cerevisiae: Gene expression and identification of novel essential genes. *Mol Biol Cell*, 15:5295–5305, 2004.

[17] T. Minka. Estimating a dirichlet distribution. Technical report, MIT, 2003.

[18] A. Niemistö, T. Aho, H. Thesleff, M. Tiainen, K. Marjanen, M. Linne, and O. Yli-Harja. Estimation of population effects in synchronized budding yeast experiments. In *Image Processing: Algorithms and Systems II*, 2003.

[19] A. Niemistö, M. Nykter, T. Aho, H. Jalovaara, K. Marjanen, M. Ahdesmäki, P. Ruusuvuori, M. Tiainen, M. Linne, and O. Yli-Harja. Distribution estimation of synchronized budding yeast population. In *Winter International Symposium on Information and Communication Technologies*, 2004.

[20] P. Nurse. A long twentieth century of the cell cycle and beyond. *Cell*, pages 71–78, 2000.

[21] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *IEEE*, 1989.

[22] S. Roweis and Z. Ghahramani. A unifying review of linear Gaussian models. *Neural Computation*, 11:305–345, 1999.

[23] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Andres, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell*, 1998.

[24] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust Monte Carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141, 2000.

siRNA Off target Search: A Hybrid q gram Based Filtering Approach*

Wenzhong Zhao
University of New Mexico
Department of Computer Science
Albuquerque, NM 87131 0001
wzhao@cs.unm.edu

Terran Lane
University of New Mexico
Department of Computer Science
Albuquerque, NM 87131 0001
terran@cs.unm.edu

ABSTRACT

Designing highly effective and gene-specific short interfering RNA (siRNA) sequences is crucial for any biological applications involving RNA interference (RNAi). A critical requirement for applying RNAi process in therapeutic applications is the ability to predict and to avoid side effect interactions with unintended transcripts (messager RNA, or mRNA). In this paper, we propose a flexible framework for detecting siRNA off-target effects. The framework can also be extended with minor changes to other applications such as selecting PCR primers or microarray nucleotide probes.

Based on the framework, we have developed and implemented a new homology sequence search program – *siRNA Off-target Search* (SOS). SOS uses a hybrid, q-gram based approach, combining two filtering techniques using overlapping and non-overlapping q-grams. This approach considers three types of imperfect matches based on biological experiments: G:U wobbles, mismatches, and bulges. The three main improvements over existing methods are: 1) introduce a more general cost model (an affine bulge cost model) for siRNA-mRNA off-target alignment; 2) use separate searches for alignments with and without bulges that enables efficient discovery of potential off-target candidates in the filtration phase; and 3) achieve better performance, in terms of speed and recall/precision, than BLAST in detecting potential siRNA off-targets.

General Terms

Algorithms, Experimentation

Keywords

RNA Interference, siRNA Off-target Search, Approximate Pattern Matching, and Sequence Alignment

1. INTRODUCTION

RNA interference (RNAi) is a recently discovered post-transcriptional gene silencing (PTGS) mechanism that seems to play both regulatory and immunological roles in the eukaryotic genetic system [1, 9, 17, 23]. RNAi has aroused a great deal of excitement in both therapeutic and genomic experimental communities because of its potential for treatment of a wide spectrum of diseases such as HIV [11]; Huntington's diseases [25]; and certain classes of cancers [3, 9], in addition to its demonstrated use in functional genomic studies via controlled gene knockdown [5, 14]. A critical requirement for the use of RNAi process in therapeutic applications is the ability to predict and to avoid side effect interactions with unintended genes. We develop a flexible siRNA off-target search program for detecting potential off-target reactions with unintended genes.

At the heart of the RNAi cleavage event is the degree of similarity between the target messager RNA (mRNA) and an initiator molecule, known as a short-interfering RNA (siRNA). By introducing a siRNA into a cell, we can induce the cellular machinery into degrading the mRNA product of a targeted gene and prevent further translation of the mRNA into protein. Thus, we can suppress the function of a specific (e.g., disease-related) gene. Early RNAi studies indicated that RNAi process is highly specific [7, 8]. However, recent experimental results strongly suggest that siRNAs with imperfect matches can still knock down unintended mRNAs with high silencing efficacy [10, 19, 21]. Three types of imperfect matches have been studied in biological experiments: mismatches [10, 20], G:U wobbles [12, 20], and internal bulges [6]. In some cases, siRNAs can tolerate several mismatches to the target sequence [10]. A recent study [13] shows that about 75% of 359 published siRNAs have a risk of non-specific effects.

Designing highly effective and specific siRNAs is crucial for therapeutic or genomic applications of the RNAi process. siRNA efficacy has been studied extensively and design rules have been established for selecting effective siRNAs (e.g., [18, 23]). However, there is an urgent need to evaluate the significance of siRNA off-target reactions with unintended sequences. Since a siRNA recognizes its targets by sequence complementarity, potential off-targets can be predicted by approximate sequence matching. However, this requires a pairwise sequence alignment between the siRNA and every gene in the genome, which can be very expensive for traditional sequence alignment algorithms such as dynamic programming.

*This work was supported by NIH Grant Number 1P20RR18754 from the Institutional Development Award Program of the National Center for Research Resources.

BIOKDD '2005 Chicago, IL USA
Copyright 2005 ACM 1-59593-213-5 ...$5.00.

Several programs that employ filtering techniques have been developed, including BLAST [2], PatternHunter [15], and QUASAR [4]. BLAST and PatternHunter filter out unrelated regions using contiguous and gapped seeds, respectively. They run much faster than dynamic programming, but both use lossy filtering techniques and thus frequently overlook off-target candidates [13]. QUASAR uses a q-gram based lossless filtering technique, but is limited to Hamming or Levenshtein distance alignments. There are also algorithms specific for siRNA selection (e.g., [13, 26]), but most deal only with mismatches.

We are interested in developing fast and lossless methods for detecting potential siRNA off-target reactions using approximate sequence matching. Our major contribution is the development and implementation of a new homology sequence search program – *siRNA Off-target Search* (SOS) – which uses a hybrid, q-gram based approach, combining two filtering techniques using overlapping and non-overlapping q-grams. The three main improvements over existing methods are:

- SOS introduces a more general cost model (an affine bulge cost model) for siRNA-mRNA off-target alignment;
- SOS uses separate searches for alignments with and without bulges that enables more efficient discovery of potential off-target candidates; and
- SOS is faster and more accurate in finding off-target candidates than BLAST, which is commonly used for siRNA off-target detection.

The rest of the paper is organized as follows. We introduce an affine bulge cost model as a measure for siRNA-mRNA off-target alignments in Section 2. In Section 3, we discuss two q-gram based filtering techniques for determining the search criteria which allow us to locate potential off-target candidates efficiently. We describe our computational experiments and report preliminary results in Section 4. Finally, we conclude and describe future work in Section 5.

2. SIRNA MRNA OFF TARGET ALIGNMENT

Consider a siRNA p, target gene g_i and mRNA $g_j \in G$, where G is the collection of genes in the genome. We define a *semi-global alignment* (alignment for short) between a siRNA and a mRNA to be a 5-tuple $A = \langle d, w, m, B_s, B_m \rangle$, where d, w, and m are the numbers of identical matches, G:U wobbles, and mismatches in the alignment; and $B_s = \{b_s\}$ and $B_m = \{b_m\}$ are the two sets of bulges on the siRNA and on the mRNA, respectively.

An affine bulge cost model is defined for computing the *alignment score.*

Definition 1. *Let $A = \langle d, w, m, B_s, B_m \rangle$ be an alignment. The affine* alignment score *for alignment A, $s(A)$, can be calculated as follows:*

$$s(A) = d\alpha + w\beta + m\gamma + \sum_{b_s \in B_s} (\rho + b_s\delta) + \sum_{b_m \in B_m} (\rho + b_m\delta),$$

where α, β, γ are the per-nucleotide scores for identity, G:U wobble, and mismatch; and ρ and δ are the scores for bulge creation and extension.

Let $N_s = \sum_{b_s \in B_s} b_s$ and $N_m = \sum_{b_m \in B_m} b_m$ be the total number of nucleotides in bulges on the siRNA and mRNA, respectively. We can rewrite the above formula as $s(A) = d\alpha + w\beta + m\gamma + (|B_s| + |B_m|)\rho + (N_s + N_m)\delta$.

We assume that $\alpha < \beta \leq \gamma$ and $0 \leq \delta \leq \rho$ hold for typical distance-based affine bulge cost models. A sample affine bulge cost model for siRNA off-target alignments is shown in Table 1. Based on the experimental results available in the literature [6, 10, 12, 20], the parameters here are manually selected to reflect the effects of different types of imperfect matches on siRNA activities in RNAi process. We use this cost model throughout the paper unless otherwise specified.

Table 1: A sample affine bulge cost model for siRNA off-target alignments

Feature	Symbol	Score
Identity	α	0
G:U wobble	β	5
Mismatch	γ	10
Bulge creation	ρ	20
Bulge extension	δ	3

The typical length for a siRNA is 19-23 nucleotides long, while mRNAs are ~2000 nucleotides long. Only a small portion of nucleotides in the mRNA contributes to an off-target alignment. The semi-global alignment for off-target detection does not penalize terminal bulges on the mRNA, but does penalize terminal bulges on the siRNA. Therefore, we define an *effective subsequence* of a mRNA in an alignment as follows:

Definition 2. *Given an alignment A between a siRNA and a mRNA, the* effective subsequence *of the mRNA in A is a portion of the contiguous nucleotides that aligns with the siRNA. The length of the effective subsequence is $L = N + N_m - N_s$, where N is the length of the siRNA.*

The following example is an illustration of semi-global off-target alignments between a siRNA and a mRNA.

Example 1. *Consider two off-target alignments between a siRNA and a mRNA, as shown in Figure 1. The off-target alignment scores for alignment A_1: $s(A_1) = 18 \cdot 0 + 0 \cdot 5 + 1 \cdot 10 + (20 + 2 \cdot 3) + 0 = 36$, and for alignment A_2: $s(A_2) = 19 \cdot 0 + 1 \cdot 5 + 1 \cdot 10 + 0 + (20 + 2 \cdot 3) = 41$. The two effective subsequences of the mRNA in alignments A_1 and A_2 are shaded, and the effective lengths for the two alignments are 19 and 23, respectively.*

Let $\mathcal{A} = \{A | A$ is an alignment between a siRNA and a mRNA$\}$ be the set of all possible alignments between the siRNA and the mRNA. The *off-target score* between them, s, is defined to be the minimum alignment score, or $s = min_{A \in \mathcal{A}} s(A)$. Currently, the off-target score is used as a first cut for siRNA off-target detection; further investigation on thermodynamic properties and target structural accessibility is underway.

3. Q GRAM BASED FILTERING

Most fast sequence alignment algorithms use a two-phase approach: a filtration phase followed by a verification phase. A *filter* is a fast algorithm that discards large portions of

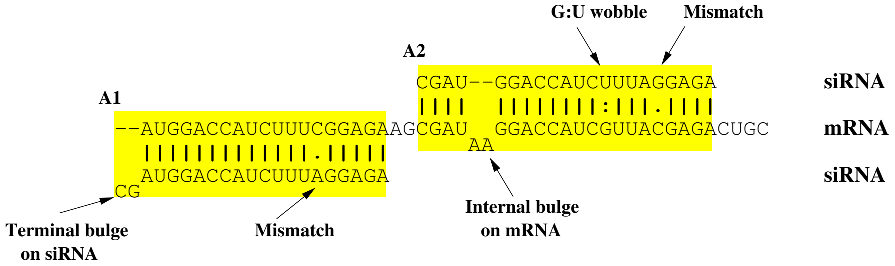


Figure 1: Semi-global alignments between a siRNA and a mRNA.

the sequence according to some *filtering criterion*, leaving the remaining part to be checked in the verification phase.

Many filters in approximate sequence matching are based on *q-grams*: substrings of length q. Given a sequence S, a *positional q-gram* q_i of S is defined to be the substring of length q that starts at position i in the sequence; $i = 1, 2, \ldots, |S| - q + 1$. The basic idea behind q-gram based filtration is that the q-gram similarity between two sequences can be captured by the number of q-gram *hits* shared by them. A *hit* is a q-gram match between two sequences:

Definition 3. *A* hit *$h(i, j)$ between a siRNA p and a mRNA g is defined to be an exact, nucleotide-for-nucleotide match between q-gram q_i in p and q-gram q_j in g.*

Alignments between a siRNA and a mRNA may or may not have bulges. The score of a bulge in a typical affine bulge cost model is higher than that of a G:U wobble or a mismatch, so alignments with bulges require more identical matches than those without bulges in order to maintain the same alignment score. Therefore, the minimum number of q-gram hits that a potential off-target candidate must share with the siRNA is higher for alignments with bulges. By separating the searches for alignments with and without bulges, we raise the *filtering criterion* for searching alignments with bulges, thus reducing the number of potential off-target candidates to be checked in the verification phase. Based on this observation, we divide all alignments between a siRNA and a mRNA into two disjoint classes: one for alignments without bulges $\mathcal{A}_\mathrm{w}$, and the other for alignments with at least one bulge $\mathcal{A}_\mathrm{b}$, and $\mathcal{A} = \mathcal{A}_\mathrm{w} \cup \mathcal{A}_\mathrm{b}$.

For a given siRNA p, we first generate positional q-grams from p, $q_1, \ldots, q_n$. We use a lookup table to locate all q-gram hits in the genome for each q-gram q_i in p. Then we use hit-processing techniques to analyze the q-gram hit lists to determine potential off-target candidates. A good way of doing that is to use a sliding window of length W, and examine q-gram hits within the region of W contiguous nucleotides in the mRNA each time. A region is considered a potential off-target candidate if there are a certain number of q-gram hits within a sliding window.

The basic procedure for searching siRNA off-targets consists of the following four phases:

1. Lookup table creation phase: Build as an indexing structure a suffix array over a sequence database G. Given the length q of q-grams, compute the start indexes of the hit lists for all q-grams in G. This step is performed once for G.

2. q-gram hit lists generation phase:

   a) Alignments without bulges: Generate non-overlapping q-grams for the selected siRNA, and search for the hit list for each q-gram.

   b) Alignments with at least one bulge: Generate overlapping q-grams for the selected siRNA, and search for the hit list for each q-gram.

3. Filtration phase:

   a) Use pigeonhole principle based approach (Corollary 1 in Section 3.1) along with hit-processing techniques to locate potential off-target candidates in G based on the q-gram hit lists from 2.a.

   b) Use q-gram lemma based approach (Corollary 2 in Section 3.2) along with hit-processing techniques to locate potential off-target candidates in G based on the q-gram hit lists from 2.b.

4. Verification phase: Potential off-target candidates from 3.a and 3.b are further checked using dynamic programming.

In the following sections, we describe the two q-gram based approaches with overlapping and non-overlapping q-grams, respectively, for determining the filtering criteria for potential off-target candidates.

3.1 Filtering based on alignments without bulges

For simplicity, we represent an alignment without bulges as $A' = \langle d', w', m' \rangle$, and the alignment score $s(A') = d'\alpha + w'\beta + m'\gamma$.

To find the *filtering criterion* for potential off-target candidates based on alignments without bulges, we apply a non-overlapping q-gram based approach that is based on the pigeonhole principle lemma.

Lemma 1. *(Pigeonhole principle lemma [16]) Let s_1 and s_2 be two sequences of the same length l with Hamming distance k. If both s_1 and s_2 are divided into $\lfloor \frac{l}{q} \rfloor$ non-overlapping q-grams, then the number of q-gram hits between s_1 and s_2 is $t_w \geq \lfloor \frac{l}{q} \rfloor - k$.*

Here we extend the pigeonhole principle lemma to the new cost model for siRNA-mRNA off-target alignments without bulges.

Lemma 2. *Let $A' = \langle d', w', m' \rangle$ be an alignment between a siRNA and a mRNA. If both the siRNA and the effective subsequence of the mRNA (i.e., with length of N in this case) are divided into $\lfloor \frac{N}{q} \rfloor$ non-overlapping q-grams, then the number of q-gram hits between them is $t_w \geq \lfloor \frac{N}{q} \rfloor - (w' + m')$, where N is the length of the siRNA.*

With respect to q-grams, a G:U wobble is the same as a mismatch, so the total Hamming distance is just $(w' + m')$.

Given an off-target threshold, the following lemma gives an upper bound for the total number of G:U wobbles and mismatches $(w' + m')$.

LEMMA 3. *Let $A' = \langle d', w', m' \rangle$ be an alignment between a siRNA and a mRNA with $s(A') \leq T$. The maximum number of G:U wobbles and mismatches in the alignment is $(m' + w') \leq \lfloor \frac{(1+\epsilon)(T-N\alpha)}{(\beta-\alpha)\epsilon+(\gamma-\alpha)} \rfloor$, where $\epsilon \geq 0$ is the ratio of the number of G:U wobbles to the number of mismatches.*[1]

Proof: $A' = \langle d', w', m' \rangle$ is an alignment without bulges, so the alignment score $s(A') = d'\alpha + w'\beta + m'\gamma$. Since $d' + w' + m' = N$, we have $d' = N - w' - m'$. Substituting d' and $w' = m'\epsilon$ into the alignment score, we have $s(A') = (N - m'\epsilon - m')\alpha + m'\epsilon\beta + m'\gamma$.

Rearranging the above equation yields $m' = \frac{s(A')-N\alpha}{(\beta-\alpha)\epsilon+(\gamma-\alpha)}$, so $m' + w' = (1+\epsilon)m' = \frac{(1+\epsilon)(s(A')-N\alpha)}{(\beta-\alpha)\epsilon+(\gamma-\alpha)}$. Since $s(A') \leq T$, we have $m' + w' \leq \frac{(1+\epsilon)(T-N\alpha)}{(\beta-\alpha)\epsilon+(\gamma-\alpha)}$. m' and w' are both integers, so $m' + w' \leq \lfloor \frac{(1+\epsilon)(T-N\alpha)}{(\beta-\alpha)\epsilon+(\gamma-\alpha)} \rfloor$. □

COROLLARY 1. *Given an alignment $A' = \langle d', w', m' \rangle$ between a siRNA and a mRNA with $s(A') \leq T$, the number of non-overlapping q-gram hits between the siRNA and the effective subsequence of the mRNA is $t_w \geq \lfloor \frac{N}{q} \rfloor - \lfloor \frac{(1+\epsilon)(T-N\alpha)}{(\beta-\alpha)\epsilon+(\gamma-\alpha)} \rfloor$.*

Corollary 1 directly follows Lemmas 2 & 3, and can be used as a *filtering criterion* for determining potential off-target candidates based on alignments without bulges.

EXAMPLE 2. *Consider an alignment $A' = \langle d', w', m' \rangle$ between a mRNA and a siRNA with length of $N = 21$ nucleotides. For a given off-target threshold T, the minimum number of q-gram hits t_w between the siRNA and the effective subsequence of the mRNA can be computed according to Corollary 1. The final results for the length of q-gram $q = 3$ are listed in Table 2.*

Table 2: The minimum number of non-overlapping q-gram hits t_w, where $N = 21$, $q = 3$, and $\epsilon = 0.2$.

Off-target threshold T	q	t_w
0	3	7
10	3	6
20	3	5
30	3	4
40	3	3
50	3	2

3.2 Filtering based on alignments with at least one bulge

To find the *filtering criterion* for potential off-target candidates based on alignments with bulges, we apply an overlapping q-gram based approach that is based on the q-gram lemma.

LEMMA 4. *(The q-gram lemma[24]) Let p be a pattern and S be a target sequence with Levenshtein distance k. The number of overlapping q-gram hits between p and S is $t_b \geq |p| - (k+1)q + 1$.*

Here we extend the *q-gram* lemma to the affine bulge cost model for siRNA-mRNA off-target alignments with bulges.

LEMMA 5. *Given an alignment $A = \langle d, w, m, B_s, B_m \rangle$ between a siRNA and a mRNA, the number of overlapping q-gram hits between the siRNA and the effective subsequence of the mRNA is $t_b \geq N - q + 1 - (w+m)q - [|B_s|(q-1) + N_s] - |B_m|(q-1)$.*

The above formula can be split into four parts. The first part, $N - q + 1$, is the total number of q-grams (or valid q-gram hits) in the siRNA. The second part means that a single mismatch or G:U wobble, in the worst case, can invalidate up to q q-gram hits. The third and fourth parts represent the maximum numbers of q-grams that can be invalidated due to bulges on the siRNA and on the mRNA, respectively.

Given an off-target threshold, the following lemma gives an upper bound for the total number of G:U wobbles and mismatches $(w + m)$.

LEMMA 6. *Let $A = \langle d, w, m, B_s, B_m \rangle$ be an alignment between a siRNA and a mRNA with $s(A) \leq T$. The maximum number of G:U wobbles and mismatches in the alignment is $(m + w) \leq \lfloor \frac{(1+\epsilon)[T-(N-N_s)\alpha-(|B|+|B'|)\rho-(N_s+N_m)\delta]}{(\beta-\alpha)\epsilon+(\gamma-\alpha)} \rfloor$, where $\epsilon \geq 0$ is the ratio of the number of G:U wobbles to the number of mismatches.*

Proof: $A = \langle d, w, m, B_s, B_m \rangle$ is an alignment, so the alignment score $s(A) = d\alpha + w\beta + m\gamma + (|B_s| + |B_m|)\rho + (N_s + N_m)\delta$. Since $d+w+m+N_s = N$, we have $d = N-w-m-N_s$. By substituting d and $w = m\epsilon$ into the alignment score, we get $s(A) = (N - m\epsilon - m - N_s)\alpha + m\epsilon\beta + m\gamma + (|B_s| + |B_m|)\rho + (N_s + N_m)\delta$.

Rearranging the above equation yields
$m = \frac{s(A)-(N-Ns)\alpha-(|B_s|+|B_m|)\rho+(N_s+N_m)\delta}{(\beta-\alpha)\epsilon+(\gamma-\alpha)}$, so
$m+w = (1+\epsilon)m = \frac{(1+\epsilon)[s(A)-(N-Ns)\alpha-(|B_s|+|B_m|)\rho+(N_s+N_m)\delta]}{(\beta-\alpha)\epsilon+(\gamma-\alpha)}$.
Since $s(A) \leq T$, we have
$m + w \leq \frac{(1+\epsilon)[T-(N-Ns)\alpha-(|B_s|+|B_m|)\rho+(N_s+N_m)\delta]}{(\beta-\alpha)\epsilon+(\gamma-\alpha)}$.
m and w are both integers, so
$m + w \leq \lfloor \frac{(1+\epsilon)[T-(N-Ns)\alpha-(|B_s|+|B_m|)\rho+(N_s+N_m)\delta]}{(\beta-\alpha)\epsilon+(\gamma-\alpha)} \rfloor$. □

COROLLARY 2. *Given an alignment $A = \langle d, w, m, B_s, B_m \rangle$ between a siRNA and a mRNA with $s(A) \leq T$, the number of overlapping q-gram hits between the siRNA and the effective subsequence of the mRNA is $t_b \geq (N - N_s) - (|B_s| + |B_m| + 1)(q-1) - \lfloor \frac{(1+\epsilon)[T-(N-Ns)\alpha-(|B_s|+|B_m|)\rho+(N_s+N_m)\delta]}{(\beta-\alpha)\epsilon+(\gamma-\alpha)} \rfloor q$.*

Corollary 2 directly follows Lemmas 5 & 6, and can be used as a *filtering criterion* for determining potential off-target candidates based on alignments with at least one bulge.

Given an affine bulge cost model, an off-target threshold T, and the length q of q-grams, the minimum number of overlapping q-gram hits t_b between a siRNA and an effective subsequence of a mRNA depends on the following four parameters: $|B_s|, |B_m|, N_s$, and N_m. Therefore, we define $(t_b)_{min}$ to be the minimum number of overlapping q-gram

[1] In this paper, we assume a uniform distribution among the four nucleotides in genomes, therefore the average value for ϵ is around 0.2. Work on more general model of G:U wobble is ongoing.

hits t_b over all possible combinations of the four parameters. $(N_s)_{max}$ and $(N_m)_{max}$ are the maximum N_s and maximum N_m, respectively, such that $t_b > 0$.

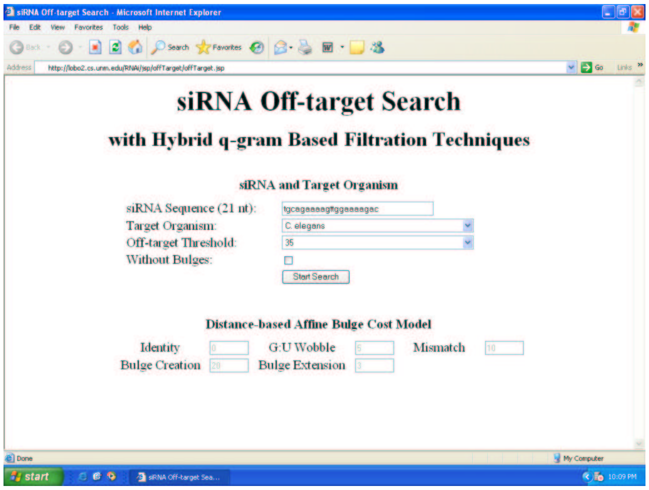
Table 3: The maximum N_s, maximum N_m, and the minimum number of overlapping q-gram hits $(t_b)_{min}$, where $N = 21$, $q = 4$, and $\epsilon = 0.2$ (Note that $(N_s)_{max}$ and $(N_m)_{max}$ need not be equal.).

Off-target threshold T	q	$(N_s)_{max}$	$(N_m)_{max}$	$(t_b)_{min}$
0, 10, 20	4	N/A	N/A	N/A
30	4	3	3	12
40	4	6	6	8
50	4	10	10	4

EXAMPLE 3. *Consider an alignment $A = \langle d, w, m, B_s, B_m \rangle$ between a mRNA and a siRNA with length of $N = 21$ nucleotides. For a given off-target threshold T, the minimum number of q-gram hits $(t_b)_{min}$ between the siRNA and the effective subsequence of the mRNA can be computed according to Corollary 2. The final results for the length of q-gram $q = 4$ are listed in Table 3, along with the $(N_s)_{max}$ and $(N_m)_{max}$.*

4. COMPUTATIONAL RESULTS

We have developed and implemented the *siRNA Off-target Search* (SOS) in Java. The online version of SOS program can be found at *http://rnai.cs.unm.edu/offTarget*. Figure 2 shows the screenshot of the SOS user interface.



siRNA Off-target Search

with Hybrid q-gram Based Filtration Techniques

siRNA and Target Organism

siRNA Sequence (21 nt): tgcagaaaagttggaaaagac
Target Organism: C. elegans
Off-target Threshold: 35
Without Bulges: □
Start Search

Distance-based Affine Bulge Cost Model

Identity	0	G:U Wobble	5	Mismatch	10
Bulge Creation	20	Bulge Extension	3		

Figure 2: A screenshot of user interface for the online version of SOS.

We performed all tests on a 3.0GHz Pentium IV machine running Linux with 1GB main memory. We applied our methods to the cDNA sequences of *C. elegans*, which contain 22,168 genes (database size: 30MB) from release WS110 of the Wormbase at Sanger Institute [22]. We first examined the performance of our SOS program, and then compared it with BLAST, a computer program commonly used for off-target detection. BLAST is downloaded from the NCBI ftp site.

Experiments were conducted to compare the number of potential off-target candidates and runtime for searching alignments with bulges with those for searching alignments without bulges. In addition, we examined the effects of separate searches for alignments with and without bulges on the overall performance of the SOS. During the experiments, we collected both the number of potential off-target candidates after the filtration phase, which is an indicator of filtration efficiency, as well as the execution time. Each experiment was repeated with 100 randomly picked siRNAs, and each data point in the figures represents the average value of the results from those tests.

Here we report the results of our computational experiments. Figure 3 compares the numbers of potential off-target candidates between searches for alignments with and without bulges. We can see that the numbers of potential off-target candidates increase with the off-target score for both cases. At lower off-target scores there are more potential off-target candidates with no bulges, while at higher off-target scores there are comparable number of potential off-target candidates for the two cases.

The number of potential off-target candidates is very low (less than 1000) at lower off-target scores, so the execution time in the filtration phase dominates. However, the number of potential off-target candidates gets much higher at higher off-target scores, so the execution time in the verification phase dominates. This is consistent with the results shown in Figure 4. At lower off-target scores the runtimes are comparable for the two cases, while at higher off-target scores the runtime for searching potential off-targets with bulges dominates. The reason is that verifying an off-target candidate with bulges is much more time-consuming than verifying an off-target candidate without bulges.

Figure 5 shows the effect of separate searches for alignments with and without bulges on the number of potential off-target candidates per siRNA. It can be seen that at lower off-target scores, separation of searches increases the potential off-target candidates, and the filtration efficiency decreases slightly. At higher off-target scores, separation of searches results in a ~90% decrease of potential off-target candidates — the filtration efficiency increases dramatically.

The number of potential off-target candidates affects the overall performance only at higher off-target scores, where the runtime of the verification phase dominates. This is supported by the fact that the total runtime with separate searches is consistently, for all off-target scores, almost one order of magnitude lower than that with no separation of searches, as shown in Figure 6.

We compared SOS with BLAST for siRNA off-target detection. SOS performs better than BLAST when matching a short sequence with a much longer sequence as in the siRNA off-target search problem. For a typical case (e.g., off-target threshold $T = 30$), SOS takes less than 0.2 second to finish the potential off-target search, as shown in Figure 6. Based on the execution time of 100 siRNA trials, BLAST takes an average of over 10 seconds for each siRNA with the default settings, which is at least one order of magnitude higher than that for SOS. Furthermore, BLAST missed a certain percentage of potential off-target sequences, as shown in Table 4. Similar results have been seen for other genes as well. Both higher off-target threshold T and longer word length w contribute towards a higher rate of undetected off-targets.

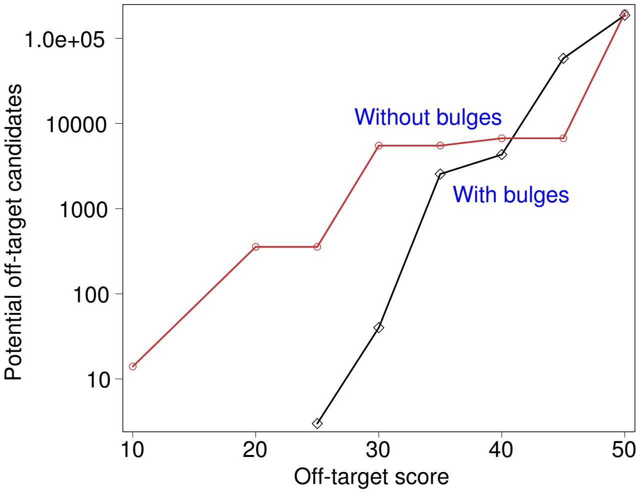


Figure 3: Number of potential off-target candidates: One searches alignments with bulges, and the other searches alignments without bulges.

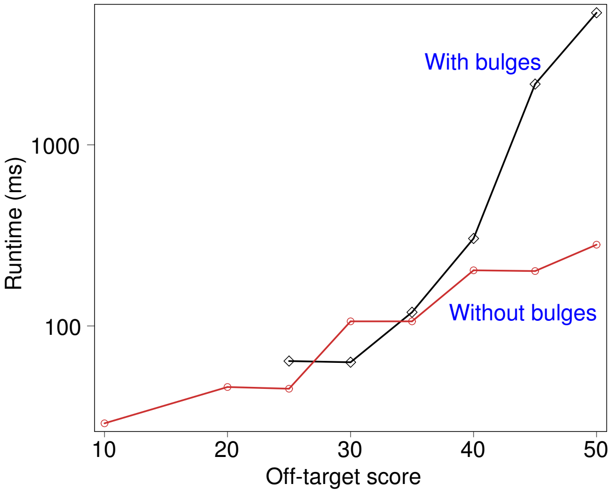


Figure 4: Runtime: One searches alignments with bulges, and the other searches alignments without bulges.

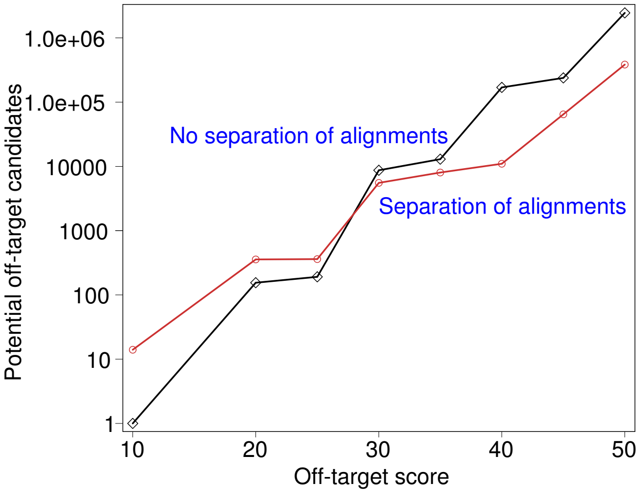


Figure 5: Total number of potential off-target candidates: One uses separate searches for alignments with and without bulges, and the other does not.

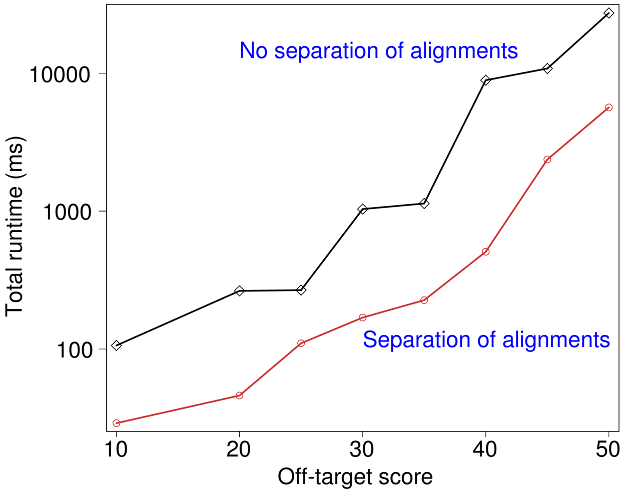


Figure 6: Total runtime: One uses separate searches for alignments with and without bulges, and the other does not.

5. CONCLUSIONS AND FUTURE WORK

We have developed and implemented the *siRNA Off-target Search* (SOS) program. It uses a hybrid, q-gram based approach, combining two filtering techniques based on overlapping and non-overlapping q-grams. This approach introduces an affine bulge cost model to measure siRNA-mRNA off-target alignment. We have demonstrated with experiments that at higher off-target scores the runtime for searching alignments with bulges dominates. By separating searches for alignments with and without bulges, we raise the filtering criterion for searching alignments with bulges, and subsequently reduce the number of potential off-target candidates to be checked in the verification phase. Therefore, using separate searches for alignments with and without bulges significantly improves the performance of the SOS. Overall, SOS achieves better performance, in terms of speed and recall/precision, than BLAST in detecting potential siRNA off-targets.

There are three major foci in our ongoing and future research: 1) Develop a specific method for G:U wobble detection in the filtration phase; 2) Use a more robust cost model considering positional information of imperfect matches; and 3) Apply gapped or partially matched q-grams in SOS.

6. REFERENCES

[1] P. Ahlquist. RNA-dependent RNA polymerases, viruses, and RNA silencing. *Science*, 296:1270–1273, May 2002.

Table 4: Percentage of potential siRNA off-targets not detected by BLAST given off-target threshold T and word length w.

Off-target threshold T	Total number of potential off-targets[a]	Percentage of off-targets not detected by BLAST[a]					
		$w = 6$	7	8	9	10	11
10	57	0.0	0.0	0.0	0.0	0.0	8.7
20	211	0.0	0.0	0.9	4.7	15.6	32.7
30	808	0.4	3.5	14.3	27.8	44.8	59.6
40	9906	3.7	15.6	32.9	50.7	65.9	76.6
50	300863	5.6	21.5	42.5	61.8	76.9	88.4

[a] These statistics are obtained based on tests for all siRNAs enumerated from a randomly picked gene (i.e., *B0024.1*, in this case) in the organism *C. elegans*.

[2] S. F. Altschul, W. Gish, W. Miller, E. Myers, and D. J. Lipman. Basic local alignment search tool. *J. Mol. Biol.*, 215:403–410, 1990.

[3] A. Borkhardt. Blocking oncogenes in malignant cells by RNA interference — new hope for a highly specific cancer treatment? *Cancer Cell*, 2(3):167–168, Sept. 2002.

[4] S. Burkhardt, A. Crauser, H. P. Lenhof, E. Rivals, P. Ferragina, and M. Vingron. q-gram based database searching using a suffix array. In *Third Annual International Conference on Computational Molecular Biology*, pages 77–83, Lyon, 1999.

[5] J.-T. Chi, H. Y. Chang, N. N. Wang, D. S. Chang, N. Dunphy, and P. O. Brown. Genomewide view of gene silencing by small interfering RNAs. *PNAS*, 100(11):6343–6346, May 2003.

[6] Y. L. Chiu and T. M. Rana. RNAi in human cells: Basic structural and functional features of small interfering RNA. *Molecular Cell*, 10:549–561, 2004.

[7] A. Dillin. The specifics of small interfering RNA specificity. *Proc. the National Academy of Sciences (PNAS)*, 100(11):6289–6291, 2003.

[8] A. Fire, S. Xu, M. K. Montgomery, S. A. Kostas, S. E. Driver, and C. C. Mello. Potent and specific genetic interference by double stranded RNA in c. elegans. *Nature*, 391:806–811, 1998.

[9] G. J. Hannon. RNA interference. *Nature*, 418:244–251, July 2002.

[10] A. L. Jackson, S. R. Bartz, J. Schelter, S. V. Kobayashi, J. Burchard, M. Mao, B. Li, G. Cavet, and P. S. Linsley. Expression profiling reveals off-target gene regulation by RNAi. *Nature Biotechnology*, 21(6):635–638, 2003.

[11] J. M. Jacque, K. Triques, and M. Stevenson. Modulation of HIV-1 replication by RNA interference. *Nature*, 418:435–438, July 2002.

[12] B. John, A. J. Enright, A. Aravin, T. Tuschl, C. Sander, and D. S. Marks. Human MicroRNA targets. *PLos Biology*, 2(11):1862–1879, 2004.

[13] O. S. Jr. and T. Holen. Many commonly used siRNAs risk off-target activity. *Biochemical and Biophysical Research Communications*, 319:256–263, 2004.

[14] R. S. Kamath, A. G. Fraser, Y. Dong, G. Poulin, R. Durbin, M. Gotta, A. Kanapin, N. Le Bot, S. Moreno, S. M., W. D. P., P. Zipperlen, and J. Ahringer. Systematic functional analysis of the caenorhabditis elegans genome using RNAi. *Nature*, 421:231–237, Jan. 2003.

[15] B. Ma, J. Tromp, and M. Li. Patternhunter: faster and more sensitive homology search. *Bioinformatics*, 18(3):440–445, 2002.

[16] E. W. Myers. A sublinear algorithm for approximate keywords searching. *Algorithmica*, 12:345–374, 1994.

[17] R. H. A. Plasterk. RNA silencing: The genome's immune system. *Science*, pages 1263–1265, May 2002.

[18] A. Reynolds, D. Leake, Q. Boese, S. Scaring, W. Marshall, and A. Khvorova. Rational siRNA design for RNA interference. *Nature Biotechnology*, 22(3):326–330, 2004.

[19] X. Z. S. P. Persengiev and M. R. Green. Nonspecific, concentration-dependent stimulation and repression of mammalian gene expression by small interfering RNAs (siRNAs). *RNA*, 10(1):12–18, 2004.

[20] S. Saxena, Z. O. Jonsson, and A. Dutta. Small RNAs with imperfect match to endogenous mRNA repress translation: implications for off-target activity of siRNA in mammalian cells. *J. Biol. Chem.*, 278(45):44312–44319, 2003.

[21] D. Semizarov, L. Frost, A. Sarthy, P. Kroeger, D. N. Halbert, and S. W. Fesik. Specificity of short interfering RNA determined through gene expression signatures. *Proc. Natl. Acad. Sci.*, 100(11):6347–6352, 2003.

[22] The Sanger Institute. From Wormbase - the C. elegans genome database. http://www.wormbase.org/, February, 2004.

[23] T. Tuschl. RNA interference and small interfering RNAs. *Chembiochem*, 2(4):239–245, 2001.

[24] E. Ukkonen. Approximate string-matching with q-grams and maximal matches. *Theoretical Computer Science*, 92:191–211, 1992.

[25] H. Xia, Q. Mao, S. L. Eliason, S. Q. Harper, I. H. Martins, H. T. Orr, H. L. Paulson, L. Yang, R. M. Kotin, and B. L. Davidson. RNAi suppresses polyglutamine-induced neurodegeneration in a model of spinocerebellar ataxia. *Nature Medicine*, 10:816–820, July 2004.

[26] T. Yamada and S. Morishita. Accelerated off-target search algorithm for siRNA. *Bioinformatics, Advance Access published on December 14, 2004*, 0(1552), 2004.

Analysis of Protein Protein Interaction Networks Using Random Walks

Tolga Can
Dept. of Computer Science
University of California
Santa Barbara, CA
tcan@cs.ucsb.edu

Orhan Çamoğlu
Dept. of Computer Science
University of California
Santa Barbara, CA
orhan@cs.ucsb.edu

Ambuj K. Singh
Dept. of Computer Science
University of California
Santa Barbara, CA
ambuj@cs.ucsb.edu

ABSTRACT

Genome wide protein networks have become reality in recent years due to high throughput methods for detecting protein interactions. Recent studies show that a networked representation of proteins provides a more accurate model of biological systems and processes compared to conventional pairwise analyses. Complementary to the availability of protein networks, various graph analysis techniques have been proposed to mine these networks for pathway discovery, function assignment, and prediction of complex membership. In this paper, we propose using random walks on graphs for the complex/pathway membership problem. We evaluate the proposed technique on three different probabilistic yeast networks using a benchmark dataset of 27 complexes from the MIPS complex catalog database and 10 pathways from the KEGG pathway database. Furthermore, we compare the proposed technique to two other existing techniques both in terms of accuracy and running time performance, thus addressing the scalability issue of such analysis techniques for the first time. Our experiments show that the random walk technique achieves similar or better accuracy with more than 1,000 times speed-up compared to the best competing technique.

Categories and Subject Descriptors

G.2.2 [**Discrete Mathematics**]: Graph Theory—*graph algorithms, network problems*; J.3 [**Life and Medical Sciences**]: Biology and Genetics

Keywords

protein networks; random walks on graphs; complex membership; pathway membership

1. INTRODUCTION

Recent developments in genome projects have shown that the complex biological functions of higher organisms are due to combinatorial interactions between their proteins. Therefore, in recent years much effort has gone into finding the complete set of interacting proteins in an organism [22]. Genome-scale protein networks have been realized with the help of high throughput methods, like yeast-two-hybrid (Y2H) [8, 19] and affinity purification with mass spectrometry (APMS) [6, 7]. However, as later studies show, the results from high throughput screens may contain significant number of false positive interactions [22]. Asthana *et al.* [1] assign probabilistic confidence values to experimentally derived interactions using the manually curated catalogs of known complexes in MIPS (Munich Information Center for Protein Sequences) [15] as a trusted reference set. In addition, information integration techniques that utilize indirect genomic evidence have provided both increased genome coverage by predicting new interactions and more accurate associations with multiple supporting evidence [4, 9, 12, 21].

Complementary to the availability of genome-scale protein networks, various graph analysis techniques have been proposed to mine these networks for pathway discovery [3, 17, 24], function assignment [11, 13, 18], and prediction of complex membership [1]. The intrinsic cluster structure of a protein network provides more accurate biological insights compared to local pairwise comparisons. Bader and Hogue [2] propose a clustering algorithm to detect densely connected regions in a protein interaction network for discovering new molecular complexes.

A biologically motivated problem is to predict new members of a partially known protein complex or pathway. In this problem, a particular *core* set of proteins is known, but the biologists are not confident that this core set is complete. The goal is to find a list of candidate proteins, preferably ranked by probability of membership in the partially known complex. As a solution to this problem, Asthana *et al.* [1] proposed a network reliability based technique to find close proximity proteins. They approximate the reliability between two nodes using Monte Carlo simulation, since the exact solution to the network reliability problem is NP-hard [20]. However, the proposed approximation technique is still computationally expensive as the number of samples for accurate reliability estimation of distant nodes can be very high. Therefore, this technique does not scale well for large protein-protein interaction networks. In this paper, as a computationally more efficient alternative, we propose

BIOKDD '05, August 2005, Chicago, Illinois, USA
Copyright 2005 ACM 1-59593-213-5 ...$5.00.

using random walks on graphs for the complex membership problem.

The random walk technique exploits the global structure of a network by simulating the behavior of a random walker [14]. The random walker starts on an initial node, i.e., the query node, and moves to a neighboring node based on the probabilities of the connecting edges. The random walker may also choose to teleport to the start node with a certain probability, called the *restart probability.* The walking process is repeated at every time tick for a certain amount of time. At the end, the percentage of time spent on a node gives a notion of its proximity to the query node. Google search engine uses a similar technique to exploit the global hyperlink structure of the Web and produce better rankings of search results [5]. Weston *et al.* [23] use the random walk technique on a protein sequence similarity graph created using PSI-BLAST scores to provide better rankings for a given query protein sequence.

The solution to the problem of finding final rankings of a random walk process can be formulated as an iterative matrix multiplication that provably converges [23]. In addition to providing a computationally much efficient alternative, the matrix formulation also allows for the random walker to start from a *set of nodes* instead of a single node. Therefore, by using the proteins of a partially known complex as the start set, the random walk technique ranks the remaining proteins in the network with respect to their proximity to the query complex. This makes the random walk technique a suitable solution for complex membership problem.

We evaluate the random walk technique on three probabilistic yeast networks using a benchmark dataset of 27 complexes from the MIPS complex catalog database [15] and 10 pathways from the KEGG [10] pathway database. Our experiments show that the ranking results provided by the random walk technique is as accurate as the network reliability technique [1] with more than 1,000 times speed-up.

The rest of the paper is organized as follows. In Section 2, we give technical details of the random walk method for the complex membership problem. In Section 3, we evaluate the proposed technique on three probabilistic yeast networks and present comparative analysis results. We conclude in Section 4.

2. METHODS

In this section, we describe the complex membership problem and present the random walk algorithm as a solution to this problem. We also discuss the competing techniques that are used in the comparative analysis.

Complex membership problem: Given a set of core proteins in a protein complex, the complex membership problem is defined as the problem of finding a set of candidate proteins, ranked according to the probability that each connects to the core complex. A good solution to this problem provides better targets for *in vivo* screening of candidate members of a protein complex. The same solution can be used for predicting candidate members of a partially known pathway if the underlying network captures functional associations as well as protein-protein interactions.

2.1 Random walks on graphs

Let $G = (V, E)$ be the graph representing a protein-protein interaction network, where V is the set of nodes (proteins), and E is the set of weighted undirected edges, where the weight shows the probability of interaction (or functional association) between protein pairs. We define the proximity of a node v to a start node s, $p_s(v)$, as follows:

```
Input: the similarity network $G = (V, E)$;
        a start node $s$;
        restart probability $c$;
Output: the proximity vector $\vec{p_s}(V)$;

Let $\vec{r_s}(V)$ be the restart vector with 0 for all its entries
        except a 1 for the entry denoted by node $s$;
Let $\mathbf{A}$ be the column normalized adjacency matrix
        defined by E;
Initialize $\vec{p_s}(V) := \vec{r_s}(V)$;
while ($\vec{p_s}(V)$ has not converged)
    $\vec{p_s}(V) := (1-c)\mathbf{A}\vec{p_s}(V) + c\vec{r_s}(V)$;
```

Figure 1: The iterative algorithm to compute the proximity of all the nodes in the graph to a given start node s.

Definition 2.1. *$p_s(v)$ is the steady state probability that a random walk starting at node s will end at node v.*

Random walk method simulates a random walker that starts on a source node, s (or a set of source nodes simultaneously). At every time tick, the walker chooses randomly among the available edges (based on edge weights), or goes back to node s with probability c. The restart probability c enforces a restriction on how far we want the random walker to get away from the start node s. In other words, if c is close to 1, the affinity vector reflects the local structure around s, and as c gets close to 0, a more global view is observed.

The probability $p_s(v)^{(t)}$, describes the probability of finding the random walker at node v at time t. The steady state probability $p_s(v)$ gives a measure of proximity to node s, and can be computed efficiently using iterative matrix operations. Figure 1 shows the iterative algorithm, which provably converges [23]. The number of iterations to converge is closely related to the restart probability c. As c gets smaller the diameter of the observed neighborhood increases, thus the number of iterations to converge gets larger. The convergence check requires the L_1-norm between consecutive $\vec{p_s}(V)$s to be less than a small threshold, e.g., 10^{-12}. In our experiments, for $c = 0.30$ the average number of iterations to converge is around 55. We give the running time performance of the random walk method for different c values in Section 3.

The details of the random walk method can be found in [14]. The main advantage of the random walk method is that it is very fast and therefore applicable to large protein networks. Another advantage is that, the method can be used to compute the proximity of a node to a set of source nodes (not just a single source node). This property is especially beneficial when a core set of members of a pathway or complex is known and the network is queried for candidate members.

2.2 Other techniques for the complex membership problem

Network reliability using Monte Carlo simulation: The solution to the two-terminal network reliability problem can be used to predict functional associations between pro-

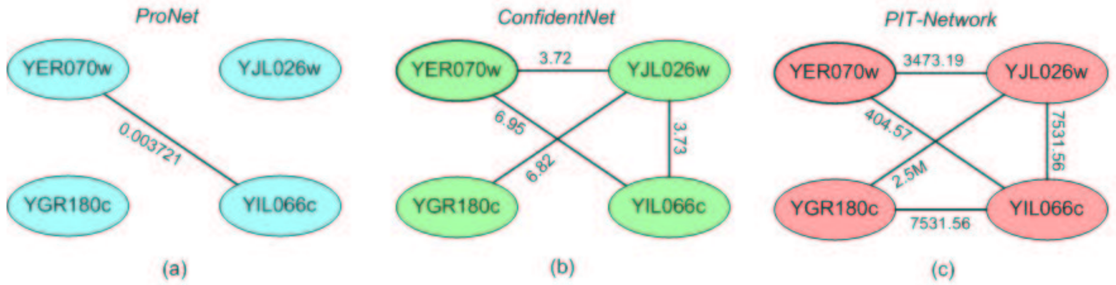


Figure 2: Associations between four members of the Ribonucleoside-diphosphate reductase complex in (a) ProNet, (b) ConfidentNet, and (c) PIT-Network. The edge weights for ProNet are probabilities with prior probability of interaction 0.007. The edge weights for ConfidentNet and PIT-Network are products of likelihoods of individual data sources. The likelihoods of these networks are not directly comparable since they are built using different number of data sources. For the network reliability technique, these likelihoods are normalized to range [0,1].

teins. In the reliability problem, we have a graph of connections between nodes in which each connection is weighted by the probability that the corresponding wire (edge) is functioning at a given time. The probability that some path of functioning wires connects the two terminals at a given time gives a measure of proximity between these terminals. The same idea can be extended to discover neighboring proteins in a protein network. The exact solution to the network reliability problem is NP-hard [20]. Monte Carlo simulation [1] is one of the approximation methods proposed for this problem. In this method, a sample of N binary networks from the probabilistic network is created according to a Bernoulli trial on each edge based on its probability. Then, breadth-first search is used to determine the existence of a path between a node in the network and the *core* complex/pathway. For each protein p in the network, the fraction F_i of sampled networks in which there exists a path between i and the core complex/pathway is counted. This process provides a ranking of all the proteins in the network. Unlike the random walk technique, this method does not normalize the incoming edges of a node when computing the *connectivity* of a protein to the core complex/pathway. The two parameters that affect the accuracy of the results and the computational efficiency of the technique are the choice of N (the number of samples) and the maximum depth for breadth-first search. In Section 3, we give accuracy and running time performance results for different values of N.

Markov random field: Markov random field method is based on belief propagation and is used to analyze protein networks by Letovsky and Kasif [13]. The method is originally proposed for function prediction but can be used to predict new members of a partially known complex or pathway. At every iteration, each node receives information about its neighbors' labels and their beliefs on the label. Each node then updates its own belief based on the distribution of its neighbors' beliefs. The updated belief is the probability of having k of M neighbors having the label. Since the belief propagation is an iterative process, nodes may mutually enhance their beliefs in the case of cycles in the network. To avoid such traps, Letovsky and Kasif propose resetting the beliefs every two iterations. The resetting is accomplished by labeling only the nodes with probability higher than some threshold (e.g., 0.8). The Markov random field method is very fast, and the underlying idea of belief propagation is very intuitive. However, there are a number of disadvantages for practical use of this method for the complex membership problem: 1) there are too many parameters to adjust, 2) no formal proof of belief bounds exist, 3) the method needs a large negative label set to suppress propagation of belief to all of the network, and 4) the result provided by the Markov random field is not a ranking but a set of nodes that are predicted to be candidate members of the core complex.

Diffusion kernels: Diffusion kernels provide a global similarity metric for the nodes of a graph. The computation of a diffusion kernel is based on the Gaussian radial basis function kernel [16, 18]. The advantages of the diffusion kernels are: 1) they are suitable for integration of multiple data sources and 2) existing kernel methods, e.g., support-vector machines, can be used for classification. The main disadvantage is that it is a measure between two nodes; therefore, a decision as to which metric should be used to compute similarity of a set of nodes to a single node (e.g., max, average, sum, etc.) is needed. The other disadvantages are: 1) computation of the diffusion kernel is expensive, 2) the only parameter β is not as intuitive as the restart probability in random walks, and 3) the effect of the edge weights on the resulting kernel is unclear. Our efforts to use diffusion kernels for the complex membership problem with default parameters were not successful as the accuracy of the results were very low compared to those of random walk, network reliability, and Markov random field techniques. Kernel methods work best with the optimum parameters whose discovery can be tedious. Therefore, we do not compare the proposed random walk method to the diffusion kernel technique.

In the next section, we evaluate the random walk technique on three probabilistic yeast networks and provide comparative results for the complex membership problem.

3. RESULTS

Many biological studies for identification of functional interactions between proteins have targeted the model organism yeast due to its small genome, extensive genetic information, and well-known biochemistry. Therefore, due to the availability of extensive experimental data, most of

Table 1: KEGG pathways used in the experiments.

KEGG pathway id:	Number of pathway members:	Pathway description:
sce00030	27	*Pentose phosphate pathway*
sce00193	30	*ATP synthesis*
sce00510	30	*N-Glycan biosynthesis*
sce00513	15	*High-mannose type N-glycan biosynthesis*
sce00600	18	*Glycosphingolipid metabolism*
sce03020	29	*RNA polymerase*
sce03022	23	*Basal transcription factors*
sce03030	21	*DNA polymerase*
sce03050	32	*Proteasome*
sce03060	10	*Protein export*

the computational studies on construction of protein networks have been on the yeast genome. Below, we describe the probabilistic yeast networks used in our experiments. The first network, ProNet (Asthana *et al.* [1]), is a probabilistic network derived from the results of four large scale experimental interaction detection techniques [6, 7, 8, 19]. ProNet contains 3,112 yeast proteins and 12,594 undirected probabilistic interactions, i.e., edges. The second network, ConfidentNet (Lee *et al.* [12]), is a probabilistic functional network of yeast genes. The associations between proteins are predicted using a Bayesian approach by combining five different information sources: mRNA coexpression, gene-fusions, phylogenetic profiles, co-citation, and protein interaction experiments. ConfidentNet contains 4,681 yeast proteins and 34,000 undirected probabilistic associations. The third network, PIT-Network (probabilistic interactome-total) (Jansen *et al.* [9]), is a combination of predicted and experimental interaction networks using a naive Bayesian approach. The predicted network is constructed using mRNA expression, GO processes, MIPS function, and essentiality data. The experimental network is constructed with the same data sources used in ProNet, but by using a fully connected Bayesian network. PIT-Network contains 2,879 yeast proteins and 24,820 interactions. To illustrate the differences between the three networks, Figure 2 shows associations between the members of a Ribonucleoside-diphosphate reductase complex in ProNet, ConfidentNet, and PIT-Network respectively.

In order to evaluate the performance of the random walk technique for the complex membership problem, we used the 27 MIPS [15] complexes examined by Asthana *et al.* [1] and 10 selected pathways from the KEGG pathway database [10]. Table 1 shows the KEGG pathways used in our experiments. We used the leave-one-out benchmark to assess the accuracy of the analysis techniques. In this benchmark, for each of the complexes and pathways examined, one member protein is left out in turn and the remaining set of member proteins is used as the core complex or the partially known pathway in a membership query. The rank of the left out protein as given by the query results provides a measure of accuracy. A successful analysis method should report the left out protein in top ranks. Therefore, in the accuracy result graphs given below, the fraction of leave-one-out queries in which the left-out protein was found above a threshold rank k is assessed.

Figure 3 and Figure 4 show the comparison results for MIPS complex queries and KEGG pathway queries on ProNet respectively. The result of the Markov random

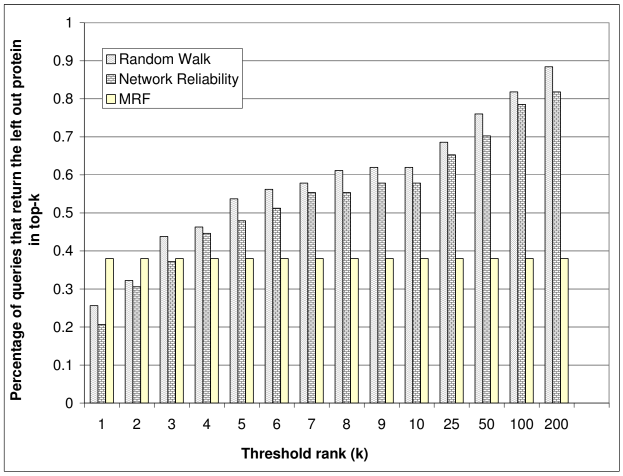


Figure 3: Comparison of analysis methods for protein complex queries on ProNet. The x-axis shows the rank threshold for the left out protein and the y-axis shows the percentage of complex queries (for a total of 121 left-out complex proteins) that the left out protein is found at (or below) the specified rank threshold.

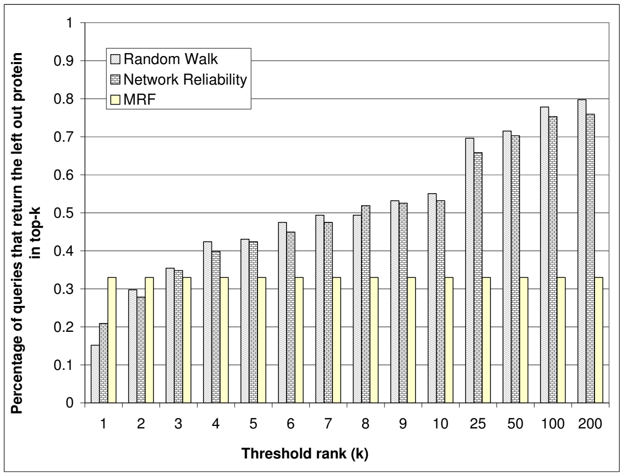


Figure 4: Comparison of analysis methods for KEGG pathway queries on ProNet.

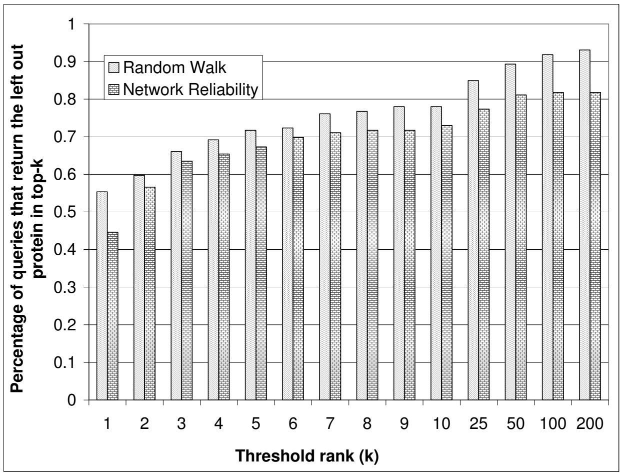


Figure 5: Comparison of random walk and network reliability techniques for MIPS complex queries on ConfidentNet.

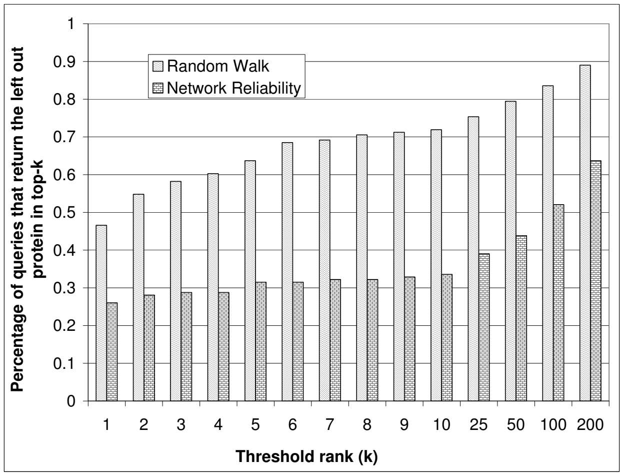


Figure 7: Comparison of random walk and network reliability techniques for MIPS complex queries on PIT-Network.

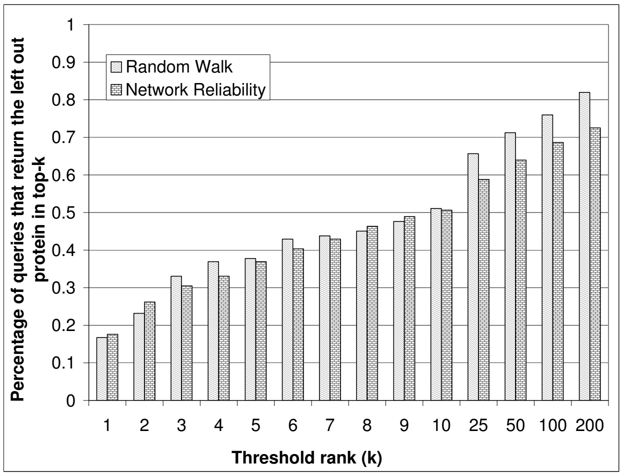


Figure 6: Comparison of random walk and network reliability techniques for KEGG pathway queries on ConfidentNet.

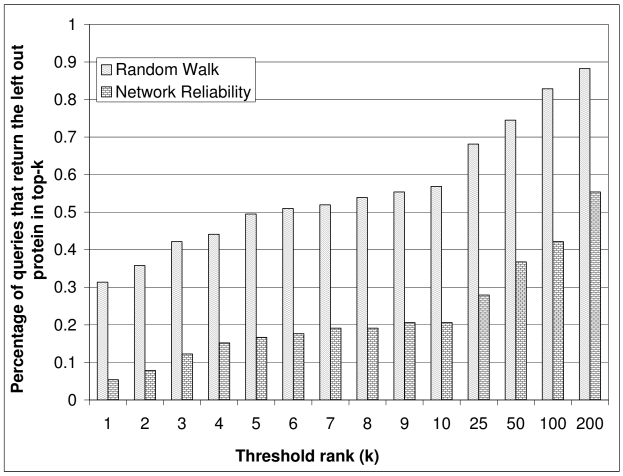


Figure 8: Comparison of random walk and network reliability techniques for KEGG pathway queries on PIT-Network.

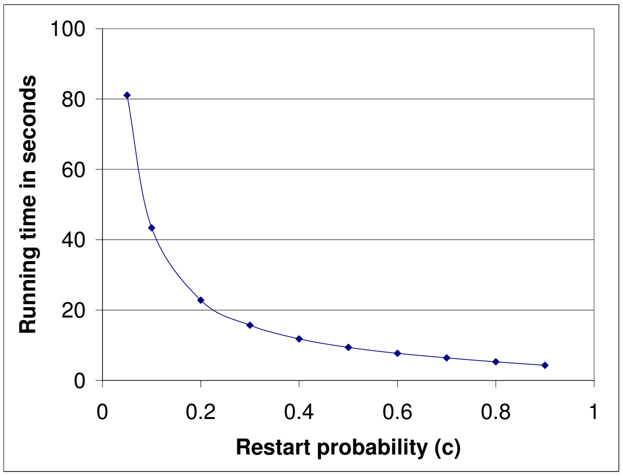


Figure 9: Running time performance of the random walk technique for varying restart probability. The queries are performed on ProNet and the time on y-axis shows the total time to complete all 121 MIPS complex leave-one out queries.

field (MRF) method is depicted as a constant height bar, because MRF method does not return a ranked list, but a set of genes predicted to be members of the complex or the pathway. The size of the set returned by the MRF method is approximately 300 for the protein networks we consider in this paper. The accuracy ratio indicates the percentage of left out proteins that are correctly predicted to be a member of the core complex/pathway. The results show that the random walk technique has similar or better accuracy compared to the network reliability technique for both complex and pathway queries. In these tests, restart probability of 0.50 was used for the random walk method and sampling size of 10,000 was used for the network reliability by Monte Carlo sampling technique. The slight decrease in the accuracy values for pathway queries is because ProNet captures only direct interactions but not functional associations.

It is clear that the accuracy of any analysis method depends also on the quality of the probabilistic network. Therefore, we performed the same benchmark tests for random walk and network reliability on ConfidentNet and PIT-Network (Figures 5 to 8). These results indicate that, regardless of the network used, random walk technique achieves similar results similar to those of the network reliability technique for the complex/pathway membership problem. One interesting observation is that the network reliability technique performs significantly worse than the random walk technique on the PIT-Network. A possible reason for this finding may be the breadth-first search threshold of 4 that is specially tuned for ProNet. The network reliability technique will perform poorly for graphs on which complex/members are placed farther apart.

Next, we analyze the effect of the restart probability for the random walk method and sample size for the Monte Carlo sampling technique (network reliability) on ProNet for MIPS complex queries. Running time behaviors of these methods on other networks are similar. Also, the running time of Markov random field method is close to that of the random walk method.

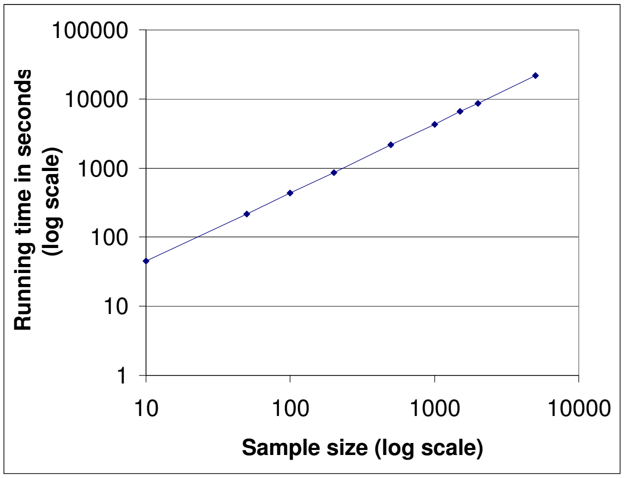


Figure 10: Running time performance of the Monte Carlo sampling approximation to the network reliability problem for varying sample size. Both axes are shown in log scale for better illustration of wide range of values. The queries are performed on ProNet and the time on y-axis shows the total time to complete all 121 MIPS complex leave-one out queries.

Figure 9 and Figure 10 show the running time performances of the random walk method and network reliability by Monte Carlo sampling method respectively. In order to compare the timing results effectively, one needs to find the optimum parameters that gives best accuracy results. Figure 11 and Figure 12 present accuracy results with respect to varying restart probability and sample size (Figure 12 is depicted as a bar graph in order show variable scale values of sample sizes more clearly). Figure 11 shows that the accuracy of the random walk technique is not sensitive to the value of restart probability. The random walk method attains the best accuracy of 54% for restart probability 0.5. On the other hand, the Monte Carlo sampling technique has the best accuracy of 51% for sample sizes 5,000 and 10,000. The running time at sample size of 5,000 is approximately 6 hours for the Monte Carlo sampling technique, whereas random walk technique achieves a better accuracy in only 9.4 seconds. This gives a speed-up of more than 2,000. Even with small sampling sizes, such as 100, where network reliability has acceptable accuracy, random walk is much faster than the Monte Carlo sampling technique, i.e. 9.4 seconds versus 437.81 seconds.

4. CONCLUSIONS

In this paper, we proposed using random walks on protein-protein interaction networks for the complex membership problem. We assessed the accuracy of the random walk technique on three different probabilistic yeast networks using a benchmark dataset of 27 complexes from the MIPS complex catalog database and 10 pathways from the KEGG pathway database. We showed that the random walk method is suitable for predicting candidate members of a core complex or partially known pathway. The most prominent property of

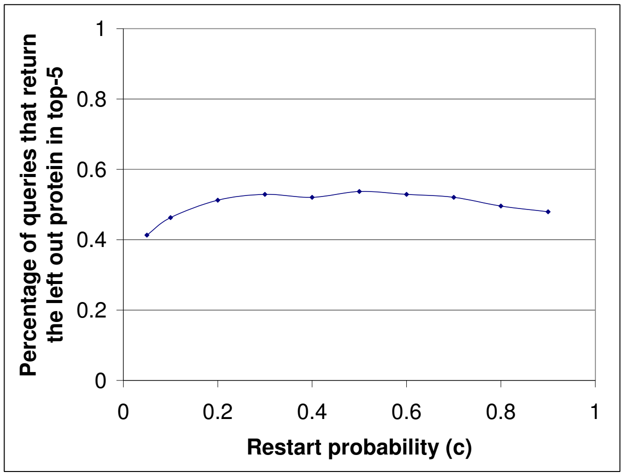


Figure 11: Accuracy of the random walk technique for varying restart probability for top-5 queries. The queries are performed on ProNet and using MIPS complexes.

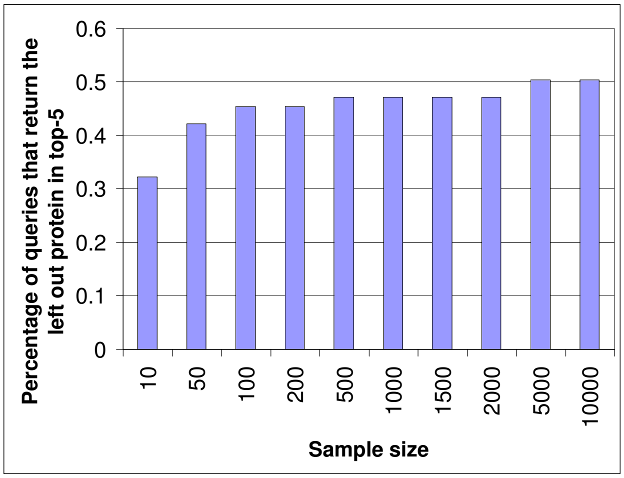


Figure 12: Accuracy of the Monte Carlo sampling technique for varying sample size for top-5 queries. The queries are performed on ProNet and using MIPS complexes.

the random walk technique is its computational efficiency. Our experiments showed that the random walk technique achieves similar or better accuracy with more than 1,000 times speed-up compared to the best competing technique. Therefore, it is a promising method that can scale well for large, genome-scale protein networks.

5. ACKNOWLEDGEMENTS

This research was supported in part by funding from the National Science Foundation: ITR-0331697, DBI-0213903, and EIA-0080134.

6. REFERENCES

[1] S. Asthana, O. D. King, F. D. Gibbons, and F. P. Roth. Predicting protein complex membership using probabilistic network reliability. *Genome Research*, 14:1170–1175, May 2004.

[2] G. D. Bader and C. W. V. Hogue. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics*, 4(2), 2003.

[3] J. S. Bader. Greedily building protein networks with confidence. *Bioinformatics*, 19(15):1869–1874, 2003.

[4] P. M. Bowers, M. Pellegrini, M. J. Thompson, J. Fierro, T. O. Yeates, and D. Eisenberg. Prolinks: a database of protein functional linkages derived from coevolution. *Genome Biology*, 5(5):R35, 2004.

[5] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30:107–117, 1998.

[6] A. C. Gavin, M. Bosche, R. Krause, P. Grandi, M. Marzioch, A. Bauer, J. Schultz, J. M. Rick, A. M. Michon, and C. M. Cruciat. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415:141–147, 2002.

[7] Y. Ho, A. Gruhler, A. Heilbut, G. D. Bader, L. Moore, S. L. Adams, A. Millar, P. Taylor, K. Bennett, and K. Boutilier. Systematic identification of protein complexes in saccharomyces cerevisiae by mass spectrometry. *Nature*, 415:180–183, 2002.

[8] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc. Natl. Acad. Sci.*, 98:4569–4574, 2001.

[9] R. Jansen, H. Yu, D. Greenbaum, Y. Kluger, N. J. Krogan, S. Chung, A. Emili, M. Snyder, J. F. Greenblatt, and M. Gerstein. A bayesian networks approach for predicting protein-protein interactions from genomic data. *Science*, 302:449–453, October 2003.

[10] M. Kanehisa, S. Goto, S. Kawashima, and A. Nakaya. The KEGG databases at GenomeNet. *Nucleic Acids Research*, 30:42–46, 2002.

[11] G. R. G. Lanckriet, M. Deng, N. Cristianini, M. I. Jordan, and W. S. Noble. Kernel-based data fusion and its application to protein function prediction in yeast. In *Proceedings of PSB*, 2004.

[12] I. Lee, S. V. Date, A. T. Adai, and E. M. Marcotte. A probabilistic functional network of yeast genes. *Science*, 306:1555–1558, November 2004.

[13] S. Letovsky and S. Kasif. Predicting protein function from protein/protein interaction data: a probabilistic approach. *Bioinformatics*, 19:i197–i204, 2003.

[14] L. Lovasz. Random walks on graphs: A survey. *Combinatorics, Paul Erdos is Eighty*, 2:353–398, 1996.

[15] H. W. Mewes, C. Amid, R. Arnold, D. Frishman, U. Guldener, G. Mannhaupt, M. Munsterkotter, P. Pagel, N. Strack, V. Stumpflen, J. Warfsmann, and A. Ruepp. MIPS: analysis and annotation of proteins from whole genomes. *Nucleic Acids Research*, 32:D41–D44, 2004.

[16] B. Schoelkopf, K. Tsuda, and J.-P. Vert, editors.

Kernel methods in computational biology. MIT Press, 2004.

[17] J. Scott, T. Ideker, R. M. Karp, and R. Sharan. Efficient algorithms for detecting signaling pathways in protein interaction networks. In *Proceedings of RECOMB*, 2005.

[18] K. Tsuda and W. S. Noble. Learning kernels from biological networks by maximizing entropy. *Bioinformatics*, 20(S1):i326–i333, 2004.

[19] P. Uetz, G. Cagney, T. A. Mansfield, R. Judson, J. R. Knight, D. Lockshon, V. Narayan, M. Srinivasan, and P. Pochart. A comprehensive analysis of protein-protein interactions in saccharomyces cerevisiae. *Nature*, 403:623–627, 2000.

[20] L. G. Valiant. The complexity of enumeration and reliability problems. *SIAM J. Comput.*, 8:410–421, 1979.

[21] C. von Mering, L. J. Jensen, B. Snel, S. D. Hooper, M. Krupp, M. Foglierini, N. Jouffre, M. A. Huynen, and P. Bork. STRING: known and predicted protein-protein associations, integrated and transferred across organisms. *Nucleic Acids Research*, 33:D433–D437, 2005.

[22] C. von Mering, R. Krause, B. Snel, M. Cornell, S. G. Oliver, S. Fields, and P. Bork. Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, 417:399–403, May 2002.

[23] J. Weston, A. Elisseeff, D. Zhou, C. S. Leslie, and W. S. Noble. Protein ranking: From local to global structure in the protein similarity network. *Proc. Nat. Acad. Sci.*, 101(17):6569–6563, 2004.

[24] Y. Yamanishi, J.-P. Vert, and M. Kanehisa. Protein network inference from multiple genomic data: a supervised approach. *Bioinformatics*, 20(S1):i363–i370, 2004.

Finding Cliques in Protein Interaction Networks via Transitive Closure of a Weighted Graph

Chris Ding, Xiaofeng He, Hanchuan Peng
Lawrence Berkeley National Laboratory
University of California, Berkeley, CA 94720
{chqding,xhe,hpeng}@lbl.gov

ABSTRACT

Finding protein functional modules in protein interaction networks amounts to finding densely connected subgraphs. Standard methods such as cliques and k-cores produce very small subgraphs due to highly sparse connections in most protein networks. Furthermore, standard methods are not applicable on weighted protein networks. We propose a method to identify cliques on weighted graphs. To overcome the sparsity problem, we introduce the concept of transitive closure on weighted graphs which is based on enforcing a transitive affinity inequality on the connection weights, and an algorithm to compute them. Using protein network from TAP-MS experiment on yeast, we discover a large number of cliques that are densely connected protein modules, with clear biological meanings as shown on Gene Ontology analysis.

Categories and Subject Descriptors

I.2 [**ARTIFICIAL INTELLIGENCE**]; I.2.6 [**Learning**]: Data mining; G.2 [**DISCRETE MATHEMATICS**]; G.2.2 [**Graph Theory**]: Cliques

1. INTRODUCTION

Proteins carry out cellular functions and processes in a modular fashion, involving multiple interacting proteins. Identification of protein functional modules becomes an urgent research topic. Fortunately, there is a large body of genome-wide comprehensive experiments on protein interaction networks. The two-hybrid genetic screen yield binary interaction data [12, 21]. Recent high throughput methods combine tagged "bait" proteins and protein-complex purification schemes with mass spectrometric measurements to yield physiologically relevant data on intact multi-protein complexes [10, 6]. Genome-wide interaction screen has been performed for several organisms, including the yeast *Saccharomyces cerevisiae* [25, 12], *vaccinia virus* [14], *hepatitis C virus* [7], and *Helicobacter pylori* [20]. However, these high-throughout experiments are often associated with large false-positives [16]. For example, interaction data obtained in two independent experiments [12, 11] and [25] only overlap less than four percent of the interactions.

A number of computational methods have been proposed for the prediction of protein interaction networks. These include gene fusion/Rosetta method [5, 13], protein sequence-based method[2], protein structure [17], phylogenetic profile [19], protein homology [8], and comparative analysis [22].

A different type method is to detect the densely connected subgraph in the protein interaction networks. The most intuitive and also simple definition of densely connected subgraph is clique, a completely connected subgraph in the protein network. One difficulty with this approach is that protein interactions are typically very sparse. Thus the cliques identified are very small. One way to overcome this problem is to use k-core [10, 1], where each protein only interacts with a fraction of other proteins in the subgraph. While this relaxes the strict definition of clique, it introduces another issue of how to choose the parameter k. Density based detection is also proposed[1, 23]. A recent approach is to use data mining association rule approach to find tightly associated proteins [27]. This approach require a transaction type data and cannot be applied directly on a graph.

The above module identification methods are on uniformly weighted (unweighted) protein interaction network where the interaction strength is either 1 or 0. This "either 1 or 0" interaction characterization is a crude description. More refined characterization would use a weighted graph, assigning a probability or level of certainty that two proteins interact. Thus the methods of identifying densely connected subgraph need to be generalized to weighted graphs.

In this paper, we address the above clique finding, network sparsity and weighted graph issues. First, we address the issue of how to define the clique in a weighted graph. We use the Motzkin-Straus theorem which relates the clique identification of an unweighted graph to the optimization of a quadratic continuous function with L_1 type linear constraints. The Motzkin-Straus approach can be generalized to weighted graphs. The key feature is that the L_1 type constraints ensure the sparsity of the solution vector, and therefore, small sizes of the resulting cliques. This definition of clique involves no parameters. (see §2).

Second, we propose a generalization of transitive closure of unweighted graphs to weighted graphs. The key idea is to show that the transitivity of similarity or affinity metric satisfies a "transitive affinity inequality" which, in some sense, is analogous to triangle inequality of distance mea-

BIOKDD '05, August 2005, Chicago, Illinois, USA
Copyright 2005 ACM 1-59593-213-5 ...$5.00.

sures. Using this transitive closure, previously less densely connected subgraph now become a "clique" and thus can be identified. This helps to resolve the sparsity problem of the protein interaction networks (see §3, §4).

In §5 and §6, we apply the proposed methods to the yeast protein interaction data, the TAP-MS experiment of multi-protein complexes. In §5, we explain how to construct weighted interaction graph from multi-protein complexes. In §6, we present the functional modules/cliques identified. The results show the advantage of transitive closure over the original sparse interaction network. The discovered protein modules have clear biological significances, as verified by Gene Ontology analysis.

2. FINDING CLIQUES IN A WEIGHTED GRAPH

We generalize the concept of clique to weighted graphs and introduce an algorithm to compute them. The generalization is based on the theorem due to Motzkin and Straus [15] which relates maximal cliques of an unweighted undirected graph to the optimization of a quadratic function.

Let $G = (V, E)$ be an unweighted undirected graph of $n = |V|$ vertices and $|E|$ edges with adjacency matrix A. Define a vector $\mathbf{x} = (x_1, \cdots, x_n)^T$ on the vertices, i.e., $\mathbf{x} \in \mathbb{R}^n$. Consider the optimization problem:

$$\max_{\mathbf{x} \in S_n} J(\mathbf{x}) = \mathbf{x}^T A \mathbf{x} \tag{1}$$

where $\mathbf{x}$ is restricted to the unit simplex S_n defined as

$$S_n: \quad x_1 + \cdots + x_n = 1,\ x_i \geq 0,\ \forall i. \tag{2}$$

The nonzero elements in the solution vector $\mathbf{x}$ play important role. In particular, we define a characteristic vector of a subset C of vertices as $\mathbf{x}^C = (x_1^c, \cdots, x_n^c)^T$, where $x_i^c = 1/|C|$ if $i \in C$, $x_i^c = 0$ otherwise. The following theorem make the connection between vertices corresponding to nonzero elements to maximal cliques.

Theorem 1[Motzkin and Straus]. (1) Subset C is the largest maximal clique if and only if its characteristic vector is a global optimal solution of the optimization problem; (2) Subset C is a maximal clique if and only if its characteristic vector is a local optimal solution of the above optimization problem.

2.1 Generalization to weighted graphs

Motzkin-Straus Theorem provides a convenient formalism for generalize the concept of cliques to weighted graphs. We propose the following definition of generalized cliques:

Generalized cliques in a weighted graph with adjacency matrix A. The subset of vertices corresponding to the nonzero elements in the optimal solution $\mathbf{x}^*$ form a clique in A.

The key to this generalization is the recognition of the L_1-type constraint of Eq.(3) in the quadratic programming problem of Eq.(1) (The L_p-norm of a vector $\mathbf{x}$ in n-dimensional space is defined as $||\mathbf{x}||_p = (\sum_{j=1}^n |x_j|^p)^{1/p},\ 1 \leq p \leq \infty$.) It is well-known (see [24, 9]) that this L_1-type constraint leads to sparse solutions, i.e., many if not most of entries in the final optimal solution $\mathbf{x}^*$ are zero. In contrast, a L_2-type constraint, such as

$$R_n: \quad x_1^2 + \cdots + x_n^2 = 1,\ x_i \geq 0,\ \forall i. \tag{3}$$

will lead to dense solution vector $\mathbf{x}^*$, i.e., most if not all elements in $\mathbf{x}^*$ are nonzero.

This sparsity property of the solution is the theoretical basis for our generalization of the Motzkin-Straus formalism to define cliques in weighted graphs.

2.2 Algorithm

The quadratic programming problem of Eq.(1) can be solved using a standard optimization package. However, there is a simple method developed in the biological evolution field [18]. The method is an iterative algorithm by updating a current solution vector using:

$$x_i \leftarrow x_i \frac{(A\mathbf{x})_i}{\mathbf{x}^T A \mathbf{x}},\ \forall i. \tag{4}$$

One can easily see the feasibility of the solution: if the initial $\mathbf{x} \in S_n$ [defined in Eq.(2)], it will remain in S_n, since $\sum_i x_i = \sum_i x_i (A\mathbf{x})_i / (\mathbf{x}^T A \mathbf{x}) = 1$. We can also prove the convergence of the algorithm, by showing that the Lagrangian function for constraint optimization

$$L(\mathbf{x}) = \mathbf{x}^T A \mathbf{x} - \lambda(\sum_i x_i - 1) \tag{5}$$

is monotonically increasing (or non-decreasing) under the above update rule: $L(\mathbf{x}^{(1)}) \leq J(\mathbf{x}^{(2)}) \leq \cdots$. The Lagrangian multiplier λ is to enforce the constraint $\sum_i x_i = 1$. Since $L(\mathbf{x})$ is bounded above, the updating algorithm converges. The proof is skipped here.

If adjacency matrix A is positive definite, the objective J is a convex function and the optimal maxima is also the global maxima. Unfortunately, for many applications, A is in generally indefinite. There are a large number of local maximas, each representing a densely connected subgraph.

We use the above updating algorithm to solve for cliques. After one local optimal solution is obtained, the clique corresponding to non-zero entries in the solution vector is extracted; these nodes are eliminated from the graph. We solve for another local optimal solution and its corresponding clique, etc. This completes the description of our graph algorithm for computing cliques in weighted graphs.

3. TRANSITIVE CLOSURE OF A WEIGHTED GRAPH

In the above, we consider densely connected subgraphs as cliques, and generalize the concept of clique on unweighted graphs to weighted graphs.

We note that the sparsity of protein interaction networks cause the cliques detected to be rather small. We seek to resolve the sparsity problem by using transitive closure idea. Here we generalize the concept of transitive closure on unweighted graphs to weighted graphs.

3.1 Transitivity and associativity

For protein networks, the weight on an edge measures the "affinity" (or interaction strength, similarity) between two proteins. We wish to study the transitivity of the affinity. Suppose we have three proteins p_i, p_j, p_k. Let w_{ij} be the interaction strength between proteins p_i, p_j. Suppose p_i interacts strongly with p_j and p_j interact strongly with p_k. With these conditions, there is a certain probability that protein p_i also interact with protein p_k.

Proteins interact in many ways: physical contact, synergystic interaction, etc. Here we make an assumption that protein interactions are transitive. In reality, the transitivity holds only approximately. For example, for 3 persons

A, B, C. If A is related to B and B is related to C, we perceive that A is likely to be related to C. The transitivity assumption is a simplification that we use to resolve the sparse interaction problem.

When the "affinity" is quantified by real nonnegative weights, $\{w_{ij}\}$, we have a number of choices to define them. If p_i interact strongly with p_j, say $w_{ij} = 0.7$ and p_j interact strongly with p_k, say $w_{jk} = 0.9$. By our transitivity assumption, protein p_i and protein p_k should also interact, i.e., w_{ik} should be significantly larger than 0, somewhere between 0.7 and 0.9. The transitive affinity may be reasonably defined in the following three ways,

$$\mathbb{T}_{max}(w_{ik}, w_{kj}) = \max(w_{ik}, w_{kj}) = 0.9, \tag{6}$$

$$\mathbb{T}_{avg}(w_{ik}, w_{kj}) = (w_{ik} + w_{kj})/2 = 0.8, \tag{7}$$

$$\mathbb{T}_{min}(w_{ik}, w_{kj}) = \min(w_{ik}, w_{kj}) = 0.7. \tag{8}$$

For the transitive affinity to be consistent along a path of (i, j, k, l), we require it has the associativity:

$$\mathbb{T}(\mathbb{T}(w_{ij}, w_{jk}), w_{kl}) = \mathbb{T}(w_{ij}, \mathbb{T}(w_{jk}, w_{kl})) \tag{9}$$

With the associativity, $\mathbb{T}(w_{ij}, w_{jk}, w_{kl})$ is uniquely defined. One can easily extend this to longer paths. It is clear that $\mathbb{T}_{avg}$ does not satisfy associativity, this rules out $\mathbb{T}_{avg}$. Both $\mathbb{T}_{max}$ and $\mathbb{T}_{min}$ satisfy associativity. Because $\mathbb{T}_{max}$ implies an very strong type of transitivity, we choose the moderate transitivity of $\mathbb{T}_{min}$. In $\mathbb{T}_{min}$, transitivity is regulated by the weakest link on the path, which is consistent with our general intuition. In the rest of this paper, we study the transitivity associated with $\mathbb{T}_{min}$. The transitive affinity on the path $(i, P, j) = (i, k_1, \cdots, k_m, j)$ is therefore

$$t_{iPj} = \min(w_{i,k_1}, w_{k_1,k_2}, \cdots, w_{k_{m-1},k_m}, w_{k_m,j}). \tag{10}$$

3.2 Maximal transitive affinity

In general, fixing vertices i, j, the transitive affinity on different paths connecting i, j are different. For this reason, we define maximal transitive affinity between i, j as

$$h_{ij} = \max_P t_{iPj}, \tag{11}$$

where P is any possible path between i, j. Given a graph, the maximal transitive affinity between any pair of vertices is uniquely defined. We can show that

$$h_{ij} \geq w_{ij}, \forall i, j. \tag{12}$$

Furthermore, we can prove that

Theorem 2. For any weighted graph, maximal transitive affinity between any pair of vertices satisfy the following transitive affinity inequality relationship

$$h_{ij} \geq \min(h_{ik},\ h_{kj}) \quad \forall i, j, k. \tag{13}$$

Now, if we replace the original weight w_{ij} by the maximal transitive affinity h_{ij}, the new graph are thus more consistent with the idea of transitivity of similarity relationship. We therefore call $\{h_{ij}\}$ as *transitive closure* of the weighted undirected graph with weights W.

Consider the case when the initial weights already satisfy the transitive affinity inequality. If we follow the above steps to compute the transitive closure, would we get something new?

Theorem 3. Suppose the initial edge weights $\{w_{ij}\}$ satisfy transitive affinity inequality. The computed maximal transitive affinity $\{h_{ij}\}$ will be identical to the initial weights: $h_{ij} = w_{ij},\ \forall i, j$.

Theorems 2 and 3, together, show that the transitive affinity inequality and maximal transitive affinity are consistent definitions.

4. TRANSITIVE AFFINITY AND ULTRA METRIC INEQUALITY

In above, we derive the transitive affinity inequality from the concept of maximal transitive affinity and show it is a consistent definition (Theorems 2 and 3).

Here we show that the transitive affinity inequality is identical to the ultra-metric of a distance metric, and therefore, a general principle. Given an affinity metric, we discuss how to compute the new weights to conform with transitive affinity inequality. The importance of these results is that the transitivity can be computed as a transitive closure and an efficient algorithm is proposed.

We recall the definition of the metric space. A function $d(x_i, x_j) = d_{ij}$ on all pairs of objects in the space, i.e., pairwise dissimilarities, is a metric, if (m1) nonnegativity: $d_{ij} \geq 0\ \forall i, j$. (m2) identity: $d_{ij} = 0$ if $x_i = x_j$. (m3) symmetry: $d_{ij} = d_{ji}$. (m4) triangle inequality

$$d_{ij} \leq d_{ik} + d_{kj}. \tag{14}$$

A metric function preserve the important notions of *distance*. Metric space has a large number of properties and useful for many problems. A special case of metric space is ultra-metric space, where the triangle inequality is replaced by a stronger ultra-metric inequality

$$d_{ij} \leq \max(d_{ik}, d_{kj}). \tag{15}$$

A dissimilarity function satisfying the ultra-metric inequality also satisfy the triangle inequality; however, not all metric functions satisfy ultra-metric inequality.

To our knowledge, so far there is no formal metric space properties based on similarity functions (instead of dissimilarity functions). Part of the reason, we believe, is that there is no clear counterpart of the triangle inequality for similarity function. However, the transitive affinity inequality we proposed in previous function is identical to the ultra-metric inequality, as we show below.

First, we note that *similarity* is an decreasing (nonincreasing) function of *distance*: the more similar the two objects are, the larger their distance is. Thus $s_{ij} = f(d_{ij})$, where $f(\cdot)$ is a monotonic decreasing function (more precisely, a nonincreasing function).

Theorem 4. The distance-based ultra-metric inequality is identical to the similarity-based transitive affinity inequality, assuming that $s_{ij} = f(d_{ij})$ is a nonincreasing function. Proof. By definition, we have

$$\begin{aligned} s_{ij} &= f(d_{ij}) \\ &\geq f(\ \max(d_{ik}, d_{kj})\ ) \qquad (16) \\ &= \min(f(d_{ik}), f(d_{kj})) \qquad (17) \end{aligned}$$

In Eq.(16), we replace d_{ij} by a possibly bigger number $\max(d_{ik}, d_{kj})$ [due to the ultra-metric inequality Eq.(15)]; and the fact that $f(\cdot)$ is nonincreasing gives the inequality. From Eq.(16) to Eq.(17), the equality is ensured since $f(\cdot)$ is nonincreasing. Substituting the definition of $s_{ij} = f(d_{ij})$, we obtain the transitive inequality $s_{ij} \geq \min(s_{ik}, s_{kj})$. □

Therefore, we can think of transitive affinity inequality as a general principle and enforce edge weights conformity with transitive affinity inequality.

Definition 5. The transitive closure of weighted graph G. For every possible (i, j, k), the edge weights are increased such that transitive affinity inequality are satisfied.

Note that the transitive closure defined above is not unique. Let $\{f_{ij}\}$ be a solution to the transitive closure problem. We may selectively increase a subset of $\{f_{ij}\}$ such that $\{f_{ij}\}$ continue to satisfy the transitive affinity inequality. Some simple example are $\{2f_{ij}\}$, $\{3f_{ij} + g_{ij}\}$, where $g_{ij} = 0$ for all edges except e_1 on which f_{ij} reach maximum and $g_{e_1} = 1$.

This non-uniqueness can be resolved by requiring the solution to be minimal, i.e., among all possible solutions $\{f_{ij}\}$, we pick the one that the net increase of edge weights

$$\Delta w^{(\alpha)} = \sum_{ij} (f_{ij}^{(\alpha)} - w_{ij})$$

is minimal. We can show that this minimal solution is unique:

Theorem 5. The minimal solution for the transitive closure is unique.

The proof Theorem 5 rely on the properties of the solutions for the transitive closure problem:

Proposition 6. Let $\{f_{ij}\}$ and $\{g_{ij}\}$ be two different solutions for the transitive closure problem. We have

$$f_{ij} \geq g_{ij}, \forall i, j. \qquad (18)$$

or the other way around:

$$f_{ij} \leq g_{ij}, \forall i, j. \qquad (19)$$

The basic reason for this property is that if for some edges, $f_{ij} \leq g_{ij}$ and edges for some other dges, $f_{ij} \geq g_{ij}$, we would have contradiction.

Therefore, the problem become finding the minimal solution for the transitive closure problem.

Clearly to verify whether a given graph weights satisfy the transitive closure, we need to go through all possible triangles to check the transitive affinity inequality. The number of all possible triangles is $\binom{n}{3} = \frac{n(n-1)(n-2)}{3!} = O(n^3)$. This is the minimal computational cost.

4.1 Modified Floyd Warshall algorithm

This transitive closure can be computed in $O(n^3)$ time by a slight modification of the well-known Floyd-Warshall algorithm for all-pair shortest paths.

Assume W is the weight matrix of a graph $G = (V, E)$ with vertex set V and edge set E. Edge e_{ij} has initial weight w_{ij}. The algorithm computes the maximal transitive affinity as the following:

```
Floyd-Warshall($W$)
$N \Leftarrow$ rows($W$)
$H \Leftarrow W$
for $k \Leftarrow 1$ to $N$
    do for $i \Leftarrow 1$ to $N$
        do for $j \Leftarrow 1$ to $N$
            $h_{ij} = \max(h_{ij}, \min(h_{ik}, h_{kj}))$
return $H$
```

The modification is on updating h_{ij}, which uses Eq.(13) for satisfying transitive affinity ity inequality.

4.2 Generalization to distance based edge weights

In §3 and §4, the weight on an edge measures the affinity (or similarity) between two nodes. These concepts and results can be equivalently generalized to the graphs where the weight on an edge measures the "distance" (or dis-similarity). Here, the "transitive distance" use $\mathbb{T}_{\max}$ in §3.1; the maximal transitive affinity in Eq.(11) is replaced by "minimal transitive distance", (close the concept of "shortest distances" between two nodes), which can be shown to satisfy the ultrametric inequality Eq.(15).

5. CONVERTING UNWEIGHTED MULTI PROTEIN COMPLEX DATA TO WEIGHTED PROTEIN INTERACTION NETWORKS

At present, two datasets summarizing high-throughput analysis of multi-protein complexes are available for the yeast *S. Cerevisiae*[6, 10]. Coupling different purification (immunoprecipitation and tandem affinity purification (TAP)) and labeling schemes with mass spectrometry (MS) both studies used bait proteins to identify physiologically intact protein complexes . Two independent studies[3, 26] showed that the TAP-MS dataset by Gavin, *et al.* [6] had the highest accuracy for predicting protein functions. Hence we have chosen this dataset.

We need to convert the protein complex data to a weighted protein interaction network (graph). A protein complex is an assembly of a small number of proteins in permanent contact and is usually perform a clear and specific biological function. A multi-protein complex can be represented as a bipartite graph. This representation allows us to infer a number of important quantities.

A bipartite graph has two type of nodes: p-nodes, $p_1, \cdots, p_m$, denoting proteins; and c-nodes, $c_1, \cdots, c_n$, denoting protein complexes A protein complex (c-node) has edges connecting to each of its constituent proteins (p-nodes) The entries of bipartite graph adjacency matrix $B = (b_{ij})$ is

$$b_{ij} = \begin{cases} 1 & \text{if protein } p_i \text{ is in protein complex } c_j \\ 0 & \text{otherwise} \end{cases} \qquad (20)$$

i.e., a protein complex is represented by a column in B, and a protein represented by a row in B. Starting from the bipartite graph, we can naturally obtain the following two weighted interaction networks [4].

Protein-Protein Interactions. The interaction strength between two proteins p_i, p_j is

$$(BB^T)_{ij} = \begin{pmatrix} \text{\# of protein complexes} \\ \text{containing both proteins } p_i, p_j \end{pmatrix}$$

Note $(BB^T)_{ii} = \sum_j b_{ij} =$ the number of protein complexes that protein p_i is involved.

Complex - Complex Associations. The interaction strength between two protein complexes c_i, c_j is

$$(B^TB)_{ij} = \begin{pmatrix} \text{\# of proteins shared by} \\ \text{protein complexes } c_i, c_j \end{pmatrix}$$

Note that $(B^TB)_{jj} = \sum_i b_{ij} =$ the number of proteins contained in the protein complex c_j.

6. APPLICATION TO PROTEIN COMPLEX DATA

We present the results of applying clique finding algorithm on original protein complex data set and on their transitive closure. We demonstrate that the cliques found are biologically meaningful based on the Gene Ontology analysis. This also provide annotations for a number of previously uncharacterized proteins.

As explained in §5, the TAP-MS dataset by Gavin, *et al.* [6] for yeast had the highest accuracy for predicting protein functions. Hence we have chosen this dataset. There are total 1,440 distinct proteins within 232 multi-protein complexes.

We use the protein - protein interactions induced by the bipartite graph as in §5. To see clearly the net effects of the transitive closure, we run the clique finding algorithm on two network weights: the original network $W = BB^T$ and its transitive closure W_{TC}. We obtain two sets of cliques C and C_{TC} as the results, corresponding to W and W_{TC} respectively. These cliques are shown in Figure 1a for the original sparse protein interaction network and in Figure 1b for the dense protein interaction network produced by the transitive closure. The obtained cliques are summarized below:

	# of cliques	Avg-Size	int-weight	ext-weight
C	296	4	0.59	0.014
C_{TC}	82	15	0.68	0.010

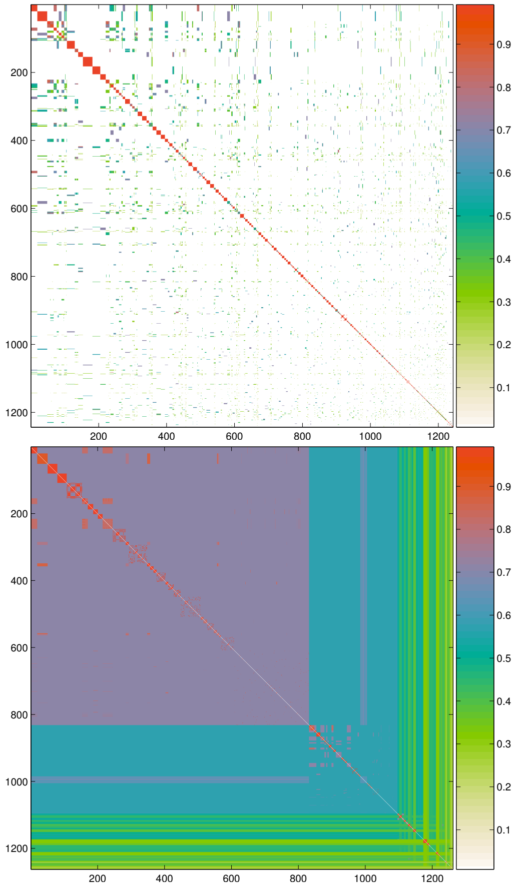


Figure 1: Weights and cliques in the weighted protein-protein interaction network. Top: original graph. Bottom: transitive closure.

On the original network, the resulting C contains 296 cliques. On the transitive closure, the resulting C_{TC} contains 82 cliques. The original network is very sparse thus we get a larger number of cliques with rather small sizes. The transitive closure is much more densely connected, thus we get a smaller number of cliques with larger sizes. These cliques also have higher average edge weight within the clique (measured in the original weight W) and lower average strength between different cliques. These indicate that proteins in the cliques detected based on the transitive closure are more densely connected with each other, meanwhile the average connectivities between different cliques are sparser. Both of these features are desirable for a protein module discovery from protein interaction networks.

We pick top 10 largest cliques and list them in Table 1. They also have large average weights on the original weight W, thus representing dense regions in the interaction network.

Clique	GO Annotation
Emg1 Imp3 Imp4 Kre31 **Mpp10** Nop14 Sof1 YMR093W YPR144C	snoRNA binding
Cus1 Msl1 Prp3 Prp9 Sme1 Smx2 Smx3 Yhc1 **YJR084W**	RNA binding
Fyv4 Mrp1 Mrp10 Mrp13 Mrp17 Mrp21 Mrp4 Mrp51 Mrps9 Nam9 Pet123 Rsm10 Rsm19 Rsm22 Rsm23 Rsm24 Rsm25 Rsm26 Rsm27 Trf4 Ubp10 YDR036C **YGR150C** YMR158W YMR188C YNL306W **YOR205C** YPL013C	structural constituent of ribosome
Atp11 **Caf130** **Caf40** Ccr4 Cdc36 Cdc39 Fas2 Not3 Not5 Pop2 Sig1 YDR214W	3'-5' -exoribonuclease activity
Apc1 Apc2 Cdc16 Cdc23 Cdc27 Doc1	ubiquitin-protein ligase & protein binding
Sec65 Srp14 Srp21 Srp54 Srp68 Srp72	signal sequence binding
Csl4 Mtr3 Rrp42 Rrp43 Rrp45 Rrp6 Ski6 Ski7	3'-5' exonuclease activity
Cft2 Fip1 Pap1 Pfs2 **Pta1** Ref2 Rna14 YGR156W Ysh1	cleavage and polyadenylation
Lsm1 Lsm2 Lsm5 Lsm6 Lsm7 Pat1 Prp24 Prp38 Snu23	RNA binding
Apl1 Apl3 Apl5 Apl6 Apm3 Apm4 Aps2 Aps3	(see Fig.3 right panel)

Table 1: Ten cliques identified by our algorithm.

Figure 2 shows the GO annotations corresponding to clique 1 (left), clique 3 (middle) and to clique 10 (right) in Table 1. Clearly the protein cliques represented by these GO annotations perform specific biological functions, namely, *snoRNA binding*, *structural constituent of ribosome*, etc. This demonstrates that these represent tightly connected cliques are meaningful protein modules.

In Fig. 3, we show an example of the effects due to transitive closure. Two smaller cliques [shown in Fig. 3 (b) and (c)] in the original interaction network W are merged into one big clique in the transitive closure W_{TC}. GO annotations show that cliques (b) and (c) have the same function as the merged clique: *structural constituent of ribosome.* This example shows why we obtain much less number of cliques on the transitive closure W_{TC}, but the sizes of these cliques are much larger. Since the larger cliques have the same function as the smaller merged ones, the cliques on the transitive closure are biologically more relevant (complete) protein modules.

Note that there are a number of uncharacterized proteins in GO (boldface protein names in Table 1). Since the GO

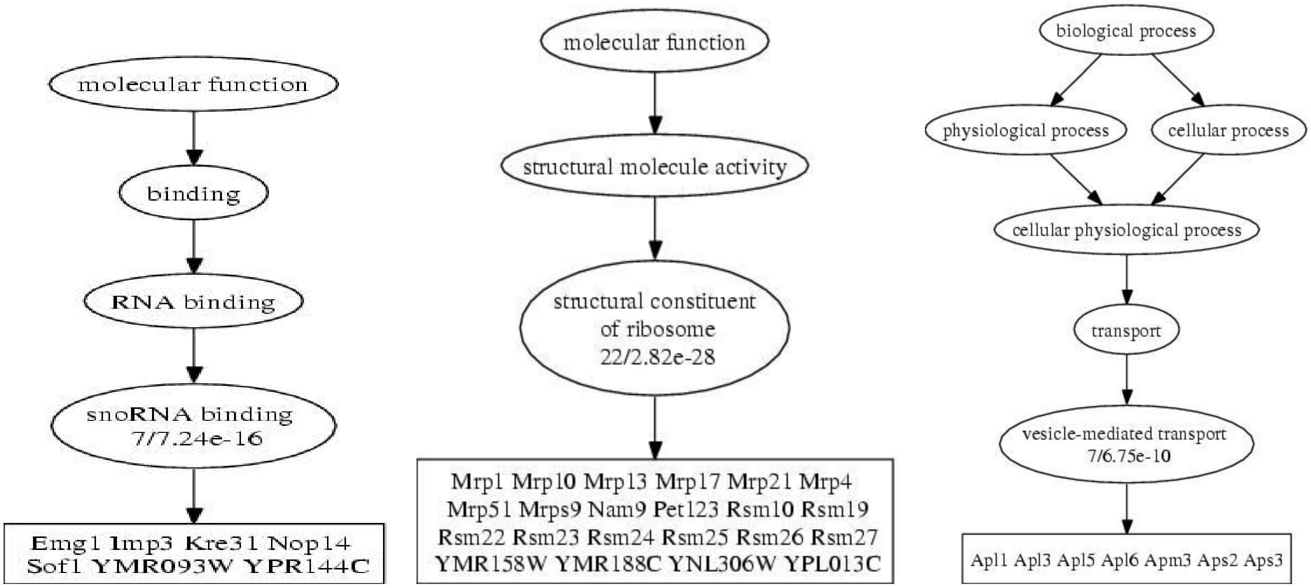

molecular function
binding
RNA binding
snoRNA binding
7/7.24e-16
Emg1 Imp3 Kre31 Nop14
Sof1 YMR093W YPR144C
molecular function
structural molecule activity
structural constituent
of ribosome
22/2.82e-28
Mrp1 Mrp10 Mrp13 Mrp17 Mrp21 Mrp4
Mrp51 Mrps9 Nam9 Pet123 Rsm10 Rsm19
Rsm22 Rsm23 Rsm24 Rsm25 Rsm26 Rsm27
YMR158W YMR188C YNL306W YPL013C
biological process
physiological process
cellular process
cellular physiological process
transport
vesicle-mediated transport
7/6.75e-10
Apl1 Apl3 Apl5 Apl6 Apm3 Aps2 Aps3

Figure 2: GO annotations for first clique in Table 1 (left), 3rd clique (middle), 10th clique (right). Proteins are listed in square box. Also shown are the number of proteins and the p-values assuming these proteins are placed there randomly.

annotation give a clear biological function for most other proteins in the clique, we infer that these uncharacterized proteins should have similar functions. Thus our approach has the additional benefit of protein annotation.

7. SUMMARY

We propose methods to identify cliques as functional modules in a non-uniformly weighted protein interaction network by (1) a generalization of Motzkin-Straus approach for identifying cliques on weighted interaction networks, and (2) generalization of transitive closure to weighted graphs, to overcome sparsity problem of protein interaction networks. All these methods are clean cut in that there are no adjustable parameters involved. Cliques detected using this approach from the yeast protein network based on the TAP-MS experiment are shown to be densely connected protein modules which have clear biological function from Gene Ontology analysis.

Acknowledgement. We thank Hui Xiong for helpful discussions regarding transitivity and Steve Holbrook on biological meaning of the cliques. This work is supported by U.S. Department of Energy, Office of Science, under contract DE-AC03-76SF00098

8. REFERENCES

[1] G. D. Bader and C. W. V. Hogue. Analyzing yeast protein-protein interaction data obtained from different sources. *Nature Biotechnology*, 2002.

[2] J. R. Bock and D. A. Gough. Predicting protein-protein interactions from primary structure. *Bioinformatics*, 17(5):455–460, 2001.

[3] M. Deng, G. Sun, and T. Chen. Assessment of the reliability of protein-protein interactions and protein function prediction. *Pacific Symposium on Biocomputing*, 2003.

[4] C. Ding, X. He, R.F. Meraz, and S.R. Holbrook. A unified representation for multi-protein complex data for modeling protein interaction networks. *Proteins: Structure, Function, and Bioinformatics*, 57:99–108, 2004.

[5] A. J. Enright, I. Iliopoulos, N. C. Kyrpides, and C. A. Ouzounis. Protein interaction maps for complete genomes based on gene fusion events. *Nature*, 203(6757):86–90, Nov 1999.

[6] A.-C. Gavin et al. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415:141–147, 2002.

[7] M. Flajolet, G. Rotondo, and L. Daviet et al. A genomic approach of the hepatitis c virus generates a protein interaction map. *Gene*, 242:369–379, 2000.

[8] N. Goffard, V. Garcia, F. Iragne, A. Groppi, and A. de Daruvar. Ippred: server for proteins interactions inference. *Bioinformatics*, 19:903–904, 2003.

[9] T. Hastie, R. Tibshirani, and J. Friedman. *Elements of Statistical Learning*. Springer Verlag, 2001.

[10] Y. Ho, A. Gruhler, A. Heilbut, et al. Systematic identification of protein complexes in saccharomyces cerevisiae by mass spectrometry. *Nature*, 2002.

[11] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki. A comprehensive two hybrid analysis to explore the yeast protein interactome. *Proc. Natl. Acad. Sci.*, 98(8):4569–4574, 2001.

[12] T. Ito, K. Tashiro, S. Muta, R. Ozawa, T. Chiba, M. Nishizawa, K. Yamamoto, S. Kuhara, and Y. Sakaki. Toward a protein-protein interaction map of the budding yeast: A comprehensive system to examine two-hybrid interactions in all possible combinations between the yeast proteins. *Proc. Natl. Acad. Sci.*, 97(3):1143–1147, 2000.

[13] E. M. Marcotte, M. Pellegrini, H.-L. Ng, D. W. Rice, T. O. Yeates, and D. Eisenberg. Detecting protein

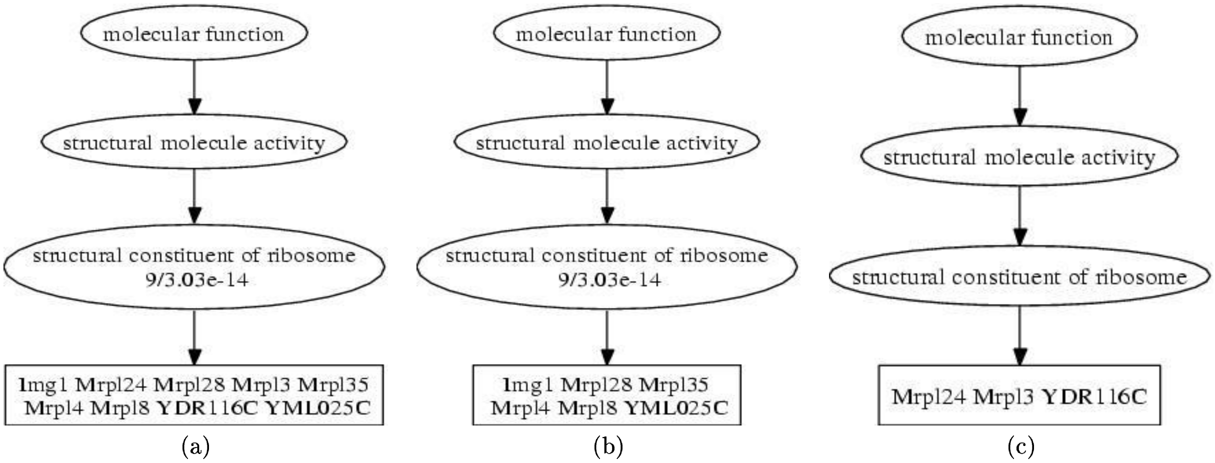

molecular function
structural molecule activity
structural constituent of ribosome
9/3.03e-14
1mg1 Mrpl24 Mrpl28 Mrpl3 Mrpl35 Mrpl4 Mrpl8 YDR116C YML025C
(a)
molecular function
structural molecule activity
structural constituent of ribosome
9/3.03e-14
1mg1 Mrpl28 Mrpl35 Mrpl4 Mrpl8 YML025C
(b)
molecular function
structural molecule activity
structural constituent of ribosome
Mrpl24 Mrpl3 YDR116C
(c)

Figure 3: GO annotations of 3 cliques. (a) The clique obtained from the transitive closure, which is formed by merging the smaller cliques shown in (b) and (c) obtained on the original network.

function and protein-protein interactions from genome sequences. *Science*, 285:751–753, 1999.

[14] S. McCraith, Ted Holtzman, Bernard Moss, and Stanley Fields. Genome-wide analysis of vaccinia virus protein-protein interactions. *Proc Natl Acad Sci USA*, 97(9):4879–4884, 2000.

[15] T.S. Motzkin and E.G. Straus. Maxima for graphs and a new proof of a theorem of turan. *Canad. J. Math.*, 17:533–540, 1965.

[16] R. Mrowka, A. Patzak, and H. Herze. Is there a bias in proteome research? *Genome Res*, 11(12):1971–1973, December 2001.

[17] J. Park, M. Lappe, and S. A. Teichmann. Mapping protein family interactions: intramolecular and intermolecular protein family interaction repertories in the pdb and yeast. *J. Mol. Biol.*, 307:929–938, 2001.

[18] M. Pelillo, K. Siddiqi, and S.W. Zucker. Matching hierarchical structures using association graphs. *IEEE. Trans. on Pattern Analysis and Machine Intelligence*, 21:1105 – 1120, 1999.

[19] M. Pellegrini, E. M. Marcotte, M. J. Thompson, D. Eisenberg, and T. O. Yeates. Assigning protein fucntions by comparative genome analysis: protein phylogenetic profiles. *Proceedings of the National Academy of Sciences (PNAS)*, 96:4285–4288, 1999.

[20] J. C. Rain, L. Selig, and H. De Reuse et al. The protein-protein interaction map of *helicobacter pylori*. *Nature*, 409:211–215, 2001.

[21] B. Schwikowski, P. Uetz, and S. Fields. A network of protein-protein interactions in yeast. *Nature Biotechnology*, 2000.

[22] R. Sharan, T. Ideker, B. Kelley, R. Shamir, and R. Karp. Identification of protein complexes by comparative analysis of yeast and bacterial protein interaction data. *Proc. Int'l Conf. Comp. Mol. Bio.(RECOMB)*, pages 282–289, 2004.

[23] V. Spirin and L.A. Mirny. Protein complexes and functional modules in molecular networks. *Proc. National Academy of Sciences*, 2003.

[24] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc B.*, 58:267–288, 1996.

[25] P. Uetz., L. Cagney, and G. Mansfield et al. A comprehensive analysis of protein-protein interactions in *saccharomyces cerevisiae*. *Nature*, 403(6770):623–627, 2000.

[26] C. von Mering, R. Krause, B. Snel, M. Cornell, S.G. Oliver, S. Fields, and P. Bork. Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, 2002.

[27] H. Xiong, X. He, C. Ding, Y. Zhang, V. Kumar, and S. R. Holbrook. Identification of functional modules in protein complexes via hyperclique pattern discovery. In *Proceedings of the Pacific Symposium on Biocomputing (PSB 2005)*, 2005.

Boosting Performance of Bio-Entity Recognition by Combining Results from Multiple Systems

Luo Si
Language Technology Inst
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
lsi@cs.cmu.edu

Tapas Kanungo
IBM Almaden Research Center
650 Harry Road
San Jose, CA 95120
kanungo@us.ibm.com

Xiangji Huang
School of Information Technology
York University
Toronto, Canada
jhuang@yorku.ca

ABSTRACT

The task of biomedical named-entity recognition is to identify technical terms in the domain of biology that are of special interest to domain experts. While numerous algorithms have been proposed for this task, biomedical named-entity recognition remains a challenging task and an active area of research, as there is still a large accuracy gap between the best algorithms for biomedical named-entity recognition and those for general newswire named-entity recognition. The reason for such discrepancy in accuracy results is generally attributed to inadequate feature representations of individual entity recognition systems and external domain knowledge.

In order to take advantage of the rich feature representations and external domain knowledge used by different systems, we propose several Meta biomedical named-entity recognition algorithms that combine recognition results of various recognition systems. The proposed algorithms – majority vote, unstructured exponential model and conditional random field – were tested on the GENIA biomedical corpus. Empirical results show that the F score can be improved from 0.72, which is attained by the best individual system, to 0.96 by our Meta entity recognition approach.

Categories & Subject Descriptors:

H.3.3 [Information Search and Retrieval]: Text Mining

General Terms: Algorithms

Keywords: Biomedical named-entity recognition; Meta recognition

1. INTRODUCTION

Biomedical literature contains a rich set of biomedical entities and information regarding the relationships and interactions among these entities. These entities and their relationships are especially useful for biologists in their quest for information [11]. The exponential growth of available biomedical literature on the Web and publicly accessible databases requires intelligent information systems that help researchers to search and analyze information. Therefore, the use of computational techniques to automatically extract useful information from biomedical texts has received increasing attention. Furthermore, to perform higher level biomedical information extraction tasks such as event extraction, summarization and question answering, most systems first identify technical terms in the domain of molecular biology that are of special interests to domain experts [11]. This is called named-entity recognition in natural language processing community [6].

The named-entity recognition task for general-purpose domain such as newswire data has been studied for a long time [3,6,22]. Both handcrafted linguistic rule based methods and machine learning based methods have been proposed for this task. Machine learning based methods [3,6] have attracted particular interest as they avoid the laborious task of manually deriving linguistic rules, and also because they can be easily adapted to new domains and new languages. Good progress has been made in named-entity recognition of newswire data and best algorithms can now achieve 'near human' performance (e.g., F score of about 0.95) [3,6,22].

The named-entity recognition task in the biomedical domain has different characteristics from that in the newswire domain. Authors tend to use more diverse notations for biomedical entities. In addition, biomedical named-entities usually have much more diverse capitalization patterns than those in newswire domain. A richer set of features, therefore, should be used to represent biomedical entities [11].

A large body of machine learning algorithms has been proposed for biomedical named-entity recognition such as hidden Markov model (HMM) [8,17,19,24,25], support vector machine (SVM) [4,13,16,19,23,25], maximum entropy markov model (MEMM) [7,14] and conditional random field (CRF) [12,15,20,23]. In order to capture the diverse characteristics of biomedical entities, different sets of features such as lexical features, affix information, orthographic features or even external resources such as gazetteers [7,25] or WWW [7,20] have been incorporated into different algorithms.

BIOKDD'05, August, 2005, Chicago, Illinois, USA.
Copyright 2005 ACM 1-59593-213-5…$5.00.

However, biomedical named-entity recognition still remains a challenging problem [11]. Despite the near-perfect performance of named-entity recognition in newswire data, similar methods do not work so well in biomedical domain and there is a large accuracy gap of about 20 points in the F score [6,9,11,25]. This problem suggests that individual biomedical named-entity systems may not cover entity representations with enough rich features and no single type of algorithm is optimal to achieve the best performance.

One natural idea of boosting performance of biomedical named-entity recognition is to combine the results of multiple biomedical entity recognition systems. This approach provides us the opportunity to combine results from multiple systems that collectively use rich and diverse feature representations and also take the advantage of utilizing multiple algorithms for achieving higher recognition accuracy.

Similar approach of combining results from multiple systems has been successfully applied in information retrieval community [1], where retrieved ranked lists from multiple information retrieval systems are combined together into a final ranked list. Empirical evidence has demonstrated that Meta retrieval approach substantially improves retrieval accuracy. However, Meta retrieval method is different from Meta entity recognition method as Meta retrieval method combines unstructured results of ranked lists while Meta entity recognition combines structured results from different named-entity recognition systems.

In this paper we propose three methods for Meta biomedical named-entity recognition. The first method uses majority vote from a set of entity recognition systems to produce combined results. This simple method does not require any training data. The second method trains an unstructured exponential model and uses the recognition results from individual systems as features to predict the correct recognition result for each word in test sentence separately. Finally, a more sophisticated structured line chain conditional random field model [12] is applied. This model utilizes structure information regarding transition among different types of entities. Although some of these techniques have been applied in other applications, to our knowledge they have never been used for Meta biomedical entity recognition.

An extensive set of empirical study has been conducted on the GENIA [1] corpus [10,11] with the task of identifying five different types of biomedical named-entities. Entity recognition results from eight different systems are considered in the Meta recognition system for combination. The best single system achieves an F score of 0.72 on the GENIA corpus [25], while the Meta recognition system with the linear chain conditional random field model achieves an F score of about 0.96. This large improvement demonstrates the power of combining multiple results for the biomedical named-entity recognition task. Furthermore, a careful comparison among different Meta recognition algorithms shows that the supervised methods of unstructured exponential model and linear conditional random field method are more effective than the simple majority vote algorithm. The structured conditional random field model achieves higher accuracy than the unstructured exponential model, which demonstrates the advantage of utilizing structure information among named-entity recognition results.

In the next section we discuss prior research related to biomedical name-entity recognition algorithms and Meta retrieval technology in information retrieval. In Section 3 we describe the three proposed Meta entity recognition algorithms --- majority vote, unstructured exponential model and structured conditional random field model. We outline the experimental methodology in Section 4 and finally present the results of our empirical study in Section 5. In Section 6 we conclude by summarizing our work and pointing out a few future research directions.

2. RELATED WORK

The approach proposed in this paper combines results from multiple biomedical named-entity recognition systems. In the next subsection we discuss specific algorithms for Bio-Entity recognition, and in the subsequent subsection we describe Meta retrieval algorithms used in information retrieval.

2.1 Algorithms for Bio-Entity Recognition

Biomedical named-entity recognition is still an active research topic, and numerous algorithms have been proposed using different feature representations. For example, in the JNLPBA [10,11] shared task of Bio-entity recognition task, eight entity recognition systems utilize different learning algorithms and different sets of features. The algorithms include variants of Support Vector Machine (SVM) [4,13,16,19,23,25], Hidden Markov Model (HMM) [8,17,19,24,25], Maximum Entropy Markov Model (MEMM) [7,14] and Conditional Random Field (CRF) Model [12,15,20,23].

Besides learning algorithms, feature representation has been recognized as a crucial factor to get good performance in Bio-Entity recognition. In the JNLPBA task [10,11], lexical features are widely used among many systems as biomedical named-entities generally have a different vocabulary from general English words. When SVM-based systems have trouble to incorporate large size of lexical features, different generalization of lexical features such as prefixes or suffixes (e.g., suffixes as ~in or ~ase for protein names) are utilized. Furthermore, some general features such as part of speech tags or word shapes as well as domain specific features such as gene sequences are also utilized in different systems. More detail can found in [11].

In addition to using features from the biomedical document itself, many systems tend to use gazetteers and other external resources for better generalization performance. Some systems use gene names from biomedical websites such as LocusLink [7] or Gene Ontology [7,13], while some other systems use the Web and construct lexicon [19,20] by collecting words that frequently appear in context with known biomedical named-entities.

To summarize, a large body of learning algorithms is available for biomedical named-entity recognition. They utilize diverse feature representations. It can be expected that the recognition results from these systems are also diverse and complementary. In the light of these facts, we believe that a good Meta biomedical named-entity recognition algorithm can take

[1] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/

advantage of the diversity of the results from multiple systems and improve the results further.

2.2 Meta Retrieval Algorithm

The approach of combining results from multiple systems has been successfully utilized in the information retrieval community [1,5,18].

Simple methods like Borda Count [1] do not require training data and favor documents that are retrieved by more individual systems against documents that are retrieved by fewer or no systems. More sophisticated algorithms that utilize training data include Naive Bayesian method [1] and logistic regression model [5]. The Naive Bayesian method makes an independence assumption among results from multiple systems, which may be inaccurate in many cases. The logistic regression model does not make the independence assumption and uses retrieved results from multiple systems as features to predict the probability of relevance for each document candidate. It has been shown that this method achieves satisfactory Meta combination results.

Although some Meta retrieval algorithms have been proposed for information retrieval, they cannot be directly used for the Meta biomedical named-entity recognition task. In particular, Meta retrieval algorithms treat only the binary case -- relevance or irrelevance of any retrieved document -- while biomedical named-entity recognition generally involves multiple types of named-entities. In addition, information retrieval systems provide unstructured ranked lists while name-entity recognition systems provide structured results of annotated sentences. These characteristics of Meta biomedical named-entity recognition task are investigated in the next section in detail.

3. ALGORITHMS FOR META BIO-ENTITY RECOGNITION

In this section, we present three algorithms for Meta biomedical named-entity recognition. All the three algorithms deal with recognition of multiple types of biomedical named-entities. The first algorithm is a simple majority vote algorithm that requires no training; the second is an unstructured exponential model that learns relative weights but does not incorporate structure information, and the third is a conditional random field model that takes full advantage of the structure information among biomedical named-entities and learns relative weights.

We now introduce the formal notation used in this paper. Let an annotated sentence be composed of words of $\vec{w}_i$ and annotated entities $\vec{s}_i$. The training data is comprised of I annotated sentences: $D=\{(\vec{w}_1,\vec{s}_1),(\vec{w}_2,\vec{s}_2)_{\ldots\ldots,}(\vec{w}_I,\vec{s}_I)\}$, where the pair $(\vec{w}_i,\vec{s}_i)$ denotes the *i*th annotated sentence. We assume that the *i*th annotated sentence contains N_i words and denote the *j*th surface word and the corresponding named-entity by (w_{ij},s_{ij}). We associate a category value for each type of named-entity and an additional "Non-entity" category for general English words. Each s_{ij} can attain any of the *K* category values. Assume that *L* annotated results are provided from *L* biomedical named-entity recognition systems. Thus, for the *i*th sentence the *l*th system's candidate results are denoted as: $\{c_{l\_i1},c_{l\_i1},\ldots\ldots,c_{l\_iN_i}\}$, where each item has a category value out of *K* choices. Finally, the task of Meta named-entity recognition algorithm is to combine the *L* candidate entity recognition results into a single result $\vec{s}_t$ for each test sentence *t*.

3.1 Simple Majority Vote Algorithm

The majority vote algorithm assumes that named-entities are correctly recognized by most individual systems, while different systems make mistakes at different places [1].

Let us introduce the binary indicator feature function $f(k,c_{l\_tj})$, which has a value 1 when the *l*th entity recognition system annotates the *j*th word in the test sentence as the entity of type *k*, and 0 when this is not true. Then the recognition rule of majority vote algorithm can be described formally as follows:

$$\hat{S}_{tj}=\arg\max_{k}\sum_{l}f(k,c_{l\_tj}) \qquad (1)$$

where *t* represents the test sentence and $\hat{S}_{tj}$ is the annotated entity result for the *j*th word in the test sentence.

One particular issue about majority vote is that votes from inaccurate entity recognition systems may not be reliable and may deteriorate the final results. Therefore, a variant of majority vote algorithm, which only considers votes from top few accurate systems, is often used in practice. This algorithm is also considered in this paper.

3.2 Unstructured Exponential Model Algorithm

One problem with the majority vote algorithm is that it treats the votes from different entity recognition systems equally. However, it is clear that more accurate systems should have more influence for the final decision than less accurate systems. The unstructured exponential model algorithm automatically derives appropriate weights for different systems from the training data, which means that those systems that are more accurate on training data are assigned with larger weights to recognize entities on test data. This type of bias is reasonable as long as the training data is representative.

Formally, the *l*th individual biomedical named-entity recognition system is associated with a weight λ_l and the probability of assigning entity of category *k* to the *j*th word in *i*th sentence is calculated as:

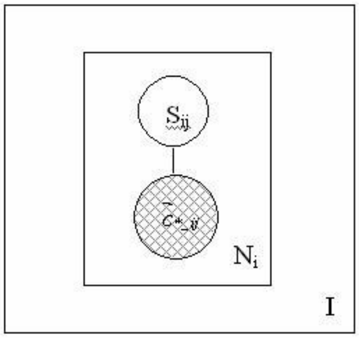


Figure 1. Graphical representation of unstructured exponential model (shared part is observed $\vec{c}_{*\_ij}$ as features from multiple entity recognition systems). Given the entity candidate features from multiple systems and model parameters, the named-entities are generated for each word separately.

$$P(\hat{S}_{ij}=k\,|\,\{w_{ij},c_{*\_ij}\})=\frac{\exp(\sum_{l}\lambda_l f(k,c_{l\_ij}))}{\sum_{k'}\exp(\sum_{l}\lambda_l f(k',c_{l\_ij}))} \qquad (2)$$

Note that no feature from the surface word itself is used in the current formulation yet. It may be useful to incorporate surface word features for more complicated combination strategy. However, empirical study in Section 5 demonstrates that this model can achieve very good performance with very limited amount of training data. Adding a lot of surface word features may cause overfitting problem with limited amount of data.

In fact, the exponential model can be seen as a multi-category extension of the logistic regression model for Meta retrieval system of information retrieval [1,5]. The graphical representation of this probabilistic model is shown in Figure 1. It can be seen from Figure 1 that given the entity features from multiple systems and model parameters, the named-entities are generated for each word separately without any interaction. That is why this model is called unstructured model.

The training criterion of this model is to maximize the conditional log-likelihood of the training data. Formally the parameter estimation problem is:

$$\vec{\lambda}^{*}=\underset{\bar{\lambda}}{\arg\max}\sum_{i,j,k}P(\hat{S}_{ij}=k)\log P(\hat{S}_{ij}=k\,|\,\{w_{ij},c_{*\_ij}\}) \qquad (3)$$

Where $P(\hat{S}_{ij}=k)$ is the empirical probability distribution for different types of named-entities of a specific word. It is 1 for one type of name-entity and zero for all the others.

The objective function in Equation (3) is a convex function and the optimization method of iterative scaling is used to obtain optimal parameter value. More detailed information about the iterative scaling method can be found in [2].

3.3 Conditional Random Field Algorithm

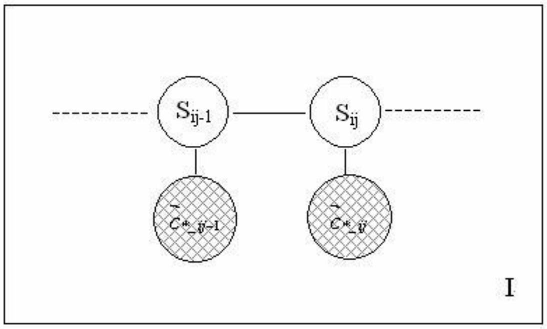


Figure 2. Graphical representation of linear conditional random field model (shadowed part is observed $\vec{c}_{*\_i*}$ as features from multiple entity recognition systems.) Given the entity candidate features from multiple systems and model parameters, the named-entities within a sentence are generated with interaction.

One important piece of useful information that is missing in the unstructured exponential model is the structure information. The named-entities assigned to nearby words are actually correlated with each other. If the previous word is recognized as a part of protein name, it is likely that the current word has a higher probability to be a part of protein entity than a cell_line entity. Conditional random field method [12] can be used to model the correlation between biomedical named-entities.

More specifically, the conditional random field model calculates conditional probabilities for whole annotated sentences instead of individual entities. In this paper, a linear chain conditional random field model is used. This is formally represented as:

$$P(\vec{s}_i=\{k_1,\ldots,k_j,\ldots,k_{N_i}\}\,|\,\vec{w}_i,c_{*\_i*}\}) = \frac{\exp(\sum_{j}\sum_{m}\lambda_m f_m(k_{j-1},k_{j,}c_{*\_i*}))}{\sum_{k_1',..k_j',...}\exp(\sum_{j}\sum_{m}\lambda_m f_m(k_{j-1}',k_j',c_{*\_i*}))} \qquad (4)$$

In particular, each feature function is associated with two concatenated entities and the corresponding candidate entity results from multiple entity recognition systems. The graphical model of linear chain conditional random field is shown in Figure 2. It can be seen that adjacent named-entities are associated with each other. This characteristic allows the conditional random field method to take advantage of structure information among entities.

The training criterion of conditional random field has a similar objective function to that of unstructured exponential model:

$$\vec{\lambda}^{*}=\underset{\bar{\lambda}}{\arg\max}\sum_{i}\log(P(\vec{s}_i\,|\,\{\vec{w}_i,c_{*\_i*}\})) \qquad (5)$$

The conditional likelihood function involves a sentence-scale normalization factor as indicated in Equation (4); the training computational complexity is much larger than that of unstructured exponential model. Quasi-Newton optimization method [21] has been shown to be more efficient than several other alternatives such as conjugate gradient and iterative scaling. This method is used in this work to train the linear chain

	Protein	DNA	RNA	Cell_type	Cell_line	All
Num of occurrences	5,067	1,056	118	1,921	500	8,662
Percent of total words	12.5%	2.6%	0.3%	4.8%	1.2%	21.4%

Table 1. Num of occurrences and percentage of total words for five types of biomedical named-entities in the corpus.

	Zho [25]	Fin [7]	Set [20]	Son [23]	Zha [24]	Rös [19]	Par [16]	Lee [13]
Recall	0.760	0.716	0.703	0.678	0.691	0.674	0.665	0.508
Precision	0.694	0.686	0.693	0.648	0.610	0.610	0.598	0.476
F-Score	0.726	0.701	0.698	0.663	0.648	0.640	0.630	0.491

Table 2. Performance of individual systems. Systems are ranked by their F scores from the highest (Left) to the lowest (Right).

conditional random field model for Meta biomedical named-entity recognition.

Given the estimated model, the recognition step of conditional random field is also more complicated than that of exponential model. A dynamic programming solution is utilized here to calculate the most likely named-entity sequence given the test sentence. Specially, a forward-backward inference algorithm like that for HMM is applied. The 'forward value' $a_j(S_{tj}=k)$ is defined as the probability of being in entity of type *k* at *j*th position given the observation up to time *j* and $\beta_j(S_{tj}=k)$ is the probability of being in entity of type *k* at *j*th position given the observation after time *j*. Recursive steps are applied to calculate the whole set of forward and backward values:

$$\begin{aligned} & a_{j+1}(S_{tj+1}=k) \\ & \qquad = \sum_{k'} a_j(k')\exp(\sum_m \lambda_m f_m(k',k,c_{*\_tj+1})) \\ & \beta_j(S_{tj}=k) \\ & \qquad = \sum_{k'} \exp(\sum_m \lambda_m f_m(k,k',c_{*\_tj+1}))\beta_{j+1}(k') \end{aligned} \tag{6}$$

Viterbi algorithm is applied with forward and backward values and finally the optimal sequence of named-entities is computed.

4. EXPERIMENTAL METHODOLOGY

We used the entity recognition results from eight different biomedical named-entity recognition systems that participated in the JNLPBA competition [2]. In the JNLPBA competition [11], each entity recognition system is required to recognize five types of entities as protein, DNA, RNA, cell_type and cell_line within documents in the GENIA corpus [10]. We utilize these results to construct Meta biomedical entity recognition system in this paper.

The recognition results are evaluated using the F score. F score is defined as: $F=(2PR)/(P+R)$, where P denotes Precision, which is the ratio of the number of correctly recognized named-entities to the number of recognized named-entities. R denotes Recall, which is the ratio of the number of correctly recognized entities to the number of true entities [11].

Since the eight systems provide results only on the test set of JNLPBA task that contains 404 documents of the GENIA corpus, we split the test data of JNLPBA into training and test data for our experiments. There are altogether 404 Medline abstracts, which are composed of 4260 sentences. The biomedical entity distribution is tabulated in Table 1. In order to fully investigate the behavior of different Meta recognition algorithms, two different training configurations were used in this work: i) 10 annotated documents for training and ii) 5 annotated documents for training. The 5 (or 10) documents that contain all the five types of annotated biomedical named-entities were randomly chosen from the 404 abstracts as training data and the remaining documents were used as test data. The training set has about 50 (or 100) sentences with about 1,250 (or 2,500) words. The random split process was repeated five times for each experiment and the evaluation results were averaged.

The performance of eight different systems on the whole corpus (404 abstracts and no training) is shown in Table 2. Three out of eight systems achieve F score around 0.7 while the F-score of other systems ranges from 0.5 to 0.65.

5. EXPERIMENTAL RESULTS

In this section we present the results of applying the proposed Meta biomedical named-entity recognition algorithms on the GENIA corpus and compare these results to individual systems. Two particular issues are investigated by the empirical study in this section:

1. Whether Meta biomedical named-entity recognition approach improves recognition accuracy over individual systems, and how do different Meta biomedical entity recognition algorithms compare against each other?
2. Detailed analysis for different types of named-entities is provided to carefully compare the results from individual systems and different Meta recognition algorithms.

5.1 Overall Recognition Accuracy

The first set of experiments was conducted to study the effectiveness of the simple majority vote algorithm. In order to show the full spectrum of its behavior, we vary the number of systems that are considered for voting. In particularly, we sort all the systems by their F scores as shown in Table 2 and use the simple majority vote algorithm to combine the results from best

[2] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/ERtask/report.html

	B1	**M_2**	**M_3**	**M_4**	**M_5**	**M_6**	**M_7**	**M_8**
Recall	0.761	0.876	0.859	0.850	0.786	0.797	0.770	0.778
Precision	0.696	0.739	0.802	0.771	0.724	0.727	0.712	0.707
F-Score	0.727	0.802	0.830	0.808	0.754	0.761	0.740	0.741

Table 3. Performance (in F score) of simple majority vote algorithms compared with the best single system (10 documents are used for training and results are averaged by five random splits). Simple majority vote algorithms combine results from different number of top systems (B1: best single system; M_2 means combination of two most accurate systems and so on).

	B1 (Baseline)	**M_8**		**M_3**		**EXP**			**CRF**		
		F Score	**Impr(%)**	**F Score**	**Impr(%)**	**F Score**	**Std**	**Impr(%)**	**F Score**	**Std**	**Impr(%)**
Recall	0.761	0.778	(+2.2%)	0.859	(+12.9%)	0.926	0.016	(+21.7%)	**0.956**	0.012	**(+25.6%)**
Precision	0.696	0.707	(+1.6%)	0.802	(+15.2%)	0.920	0.021	(+32.2%)	**0.971**	0.010	**(+39.5%)**
F-Score	0.727	0.741	(+1.9%)	0.830	(+14.2%)	0.923	0.015	(+27.0%)	**0.964**	0.011	**(+32.6%)**

Table 4. Performance of Meta biomedical named-entity systems compared with the best single system (10 documents are used for training and results are averaged by five random splits; F Score: F measure; Std: standard deviation across 5 random splits; Impr(%): Relative improvement over baseline). B1: Best single system; M_8: majority vote from eight systems; M_3: majority vote from best three systems; EXP: unstructured exponential model: CRF: conditional random field. (Standard deviation of M_8 and M_3 are not reported as they are very small)

	B1 (Baseline)	**M_8**		**M_3**		**EXP**			**CRF**		
		F Score	**Impr(%)**	**F Score**	**Impr(%)**	**F Score**	**Std**	**Impr(%)**	**F Score**	**Std**	**Impr(%)**
Recall	0.759	0.777	(+2.4%)	0.858	(+13.0%)	0.907	0.026	(+19.4%)	**0.921**	0.024	**(+21.3%)**
Precision	0.694	0.706	(+1.7%)	0.801	(+15.4%)	0.879	0.035	(+26.7%)	**0.953**	0.018	**(+37.3%)**
F-Score	0.725	0.740	(+2.0%)	0.829	(+14.3%)	0.893	0.030	(+23.3%)	**0.937**	0.021	**(+29.2%)**

Table 5. Performance of Meta biomedical named-entity systems compared with the best single system (5 documents are used for training and results are averaged by five random splits; F Score: F measure; Std: standard deviation across 5 random splits; Impr(%): Percentage improvement over baseline). Algorithm descriptions are the same as the above.

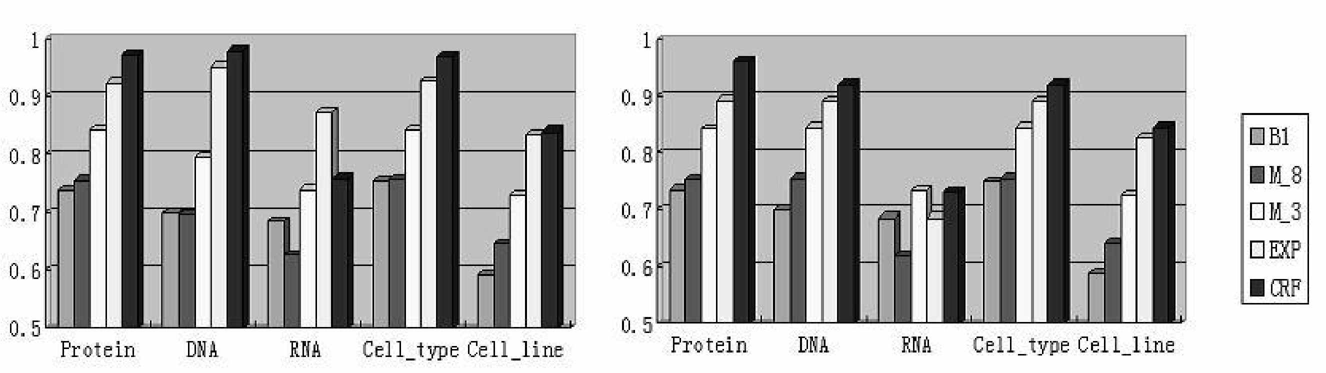
two systems (M_2), best three systems (M_3) and so on. The detailed experiments are shown in Table 3. While the majority vote algorithm does not have to be trained, we made the experimental setup identical to that used for the trainable Meta algorithms to make the evaluation results comparable: 10 documents were held for training in each of the five random splits and the remaining 394 documents were used for test (the results when 5 documents were used for training are almost identical with these results and are not shown). The majority voting algorithms did not use the 10 (and 5) training documents – only the trainable algorithm made use of them.

Note a particular issue of simple majority vote algorithm is tie breaking. If the votes from multiple systems are the same for some entities, the preference is given in the order to protein, DNA, RNA, cell_ type, cell_line and "Non-entity".

It can be seen from Table 3 that simple majority vote algorithm does achieve more accurate result than single best system. However, its performance varies with the number of systems of combination. The best results are achieved when top three or four systems are considered for voting and the accuracy drops significantly when more and more low accuracy systems are added into the combination. This behavior suggests that appropriate weights should be assigned to individual systems in order to achieve optimal performance of Meta named-entity recognition; and this is exactly the goal of the unstructured exponential model and conditional random field model

More experiments were conducted to study four types of Meta biomedical entity recognition algorithms. The algorithms are: M_8 (majority vote algorithm form all of the eight individual systems); M_3 (majority vote algorithm from three most accurate individual systems as Zho [25], Fin [7] and Set [20]); EXP (unstructured exponential model) and CRF (conditional random field model). Both the EXP and CRF algorithms take advantage of training data. Table 4 shows the results when 10 documents were available for training. It can be seen that EXP and CRF achieve a significant improvement over the best single system and also are much more accurate than the simple majority algorithm. More careful analysis shows that EXP and CRF algorithms automatically assign appropriate weights for individual systems. For example, EXP assigns more weights to the top three systems than the other systems. Furthermore, CRF algorithm generates more accurate results than the EXP algorithm. This demonstrates the power of utilizing the structure information among entities.

Another set of experiments was designed to test the behavior of different Meta entity recognition algorithms with more limited amount of training data. The experiments shown in Table 5 use only 5 documents as training data. It can be seen from Table 5 that the performance of M_8 and M_3 algorithms remain at



(a) (b)

Figure 3. Performance of best single systems and Meta recognition algorithms for different types of biomedical named entities. (a) is the case with 10 documents for training while (b) is the case with 5 documents for training. For B1, system from Zho [21] is used to predict protein, DNA, cell_type and cell_line while the system from Fin [5] is used to predict RNA.

about the same level as those in Table 4 since these algorithms do not utilize training data and their accuracy does not depend on the size of training data. The accuracy of EXP and CRF algorithms drops slightly with more limited amount of training data. However, their advantage over best single system or simple majority vote recognition algorithm is still very large. This set of experiments suggests that Meta biomedical named-entity recognition algorithms can acquire very accurate results even with very limited amount of training data (i.e., about 50 training sentences).

Other configurations with more training data have also been studied. When 15, 20 or more documents are used for training, the accuracy of EXP and CRF methods increase. However, the improvement over the results of less training data (i.e., 5 or 10 documents.) is small due to the high performance of EXP and CRF methods with limited amount of training data.

Both unstructured exponential algorithm and conditional random algorithm are very efficient. They are implemented using Matlab. It takes about 30 seconds to train the exponential model and about 2 minutes to train the conditional random field model in the case of 10 training documents. It only takes about 30 seconds for CRF to generate combined results for 394 documents while several seconds for the exponential model.

5.2 Recognition Accuracy for Different Types of Biomedical Named Entities

This set of experiments shows how Meta entity recognition algorithms improve the recognition accuracy for each type of biomedical named entity.

Figure 3 shows the performance of best single system and Meta recognition algorithms for different types of biomedical named-entities. Note that different individual systems may be optimal for different types of biomedical named-entities. For example, the system by Fin [7] has a better performance for RNA entities than the system by Zho [25]. More detail can be found in [11].

It can be seen from Figure 3 that Meta recognition algorithms CRF, EXP and M_3 achieve better performance than single best system. Unstructured exponential model and conditional random field model achieve better result than other algorithms in most cases by assigning appropriate weights to the results from multiple systems. In fact, the weights of different systems are also varied for the recognition of different types of entities. Furthermore, the CRF method provides the most accurate results in most cases, which again demonstrates the power of utilizing structure information.

6. CONCLUSION AND FUTURE WORK

Due to the large vocabulary and very diverse notations of biomedical entities, the performance of current biomedical named-entity recognition systems is still not satisfactory. Possible reasons are inadequate feature representations of individual systems and ineffectiveness of individual algorithms.

This paper proposes a Meta biomedical named-entity recognition approach by combining results from multiple systems. Three types of Meta recognition algorithms are proposed. Empirical study shows that Meta biomedical named-entity methods can substantially improve recognition accuracy over individual systems. The best results are obtained with a conditional random field method that takes the advantage of structure information for recognition. With a small amount of training data, this method provides recognition results with an F score of 0.96 while the F score of the best single system is only 0.72 [11,25]

As more and more trainable biomedical named-entity systems are available, we will apply the Meta entity recognition approach on other biomedical corpus for more complete evaluation. Training data can be used to train both individual named-entity recognition systems and the Meta recognition system. Furthermore, more sophisticated model which considers surface word features to combine results will be investigated in future work.

REFERENCES

[1] J. A. Aslam and M. Montague (2001). Models for Metasearch. In *Proceedings of the 24th Annual*

International ACM SIGIR Conference on Research and Development in Information Retrieval.

[2] A. Berger. (1997). A gentle introduction to iterative scaling. http://www-2.cs.cmu.edu/~aberger/maxent.html

[3] D. M. Bikel, R. L. Schwartz and R. M. Weischedel. (1999). An algorithm that learns what's in a name. *Machine Learning*, vol. 34, no. 1-3, pp. 211-231, 1999.

[4] Christopher J.C. Burges. (1998) A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery* 2, 121-167.

[5] A. Le Calvé, J. Savoy (2000): Database Merging Strategy Based on Logistic Regression. *Information Processing & Management,* 36(3), 341-359.

[6] DARPA. (1995). *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, Columbia, MD, USA, November. Morgan Kaufmann.

[7] J. Finkel, S. Dingare, H. Nguyen, M. Nissim, G. Sinclair and C. Manning. (2004). Exploiting Context for Biomedical Entity Recognition: From Syntax to the Web. In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004)*, Geneva, Switzerland.

[8] T. Kanungo. HMM software learning toolkit. University of Maryland Institute for Advanced Computer Studies, http://www.cfar.umd.edu/~kanungo/software/software.html

[9] J. D. Kim, T. Ohta, Y. Tateisi and J. Tsujii. (2002). Corpus-Based Approach to Biological Entity Recognition. In *Proceedings of the Second Meeting of the Special Interest Group on Test Data Mining of ISMB (BioLink-2002)*, Edmonton, Canada.

[10] J. D. Kim, T Ohta, Y. Tateishi and J. Tsujii. (2003). GENIA corpus - a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19 (Suppl.1): 180-182.

[11] J. D. Kim, T Ohta, Y. Tateishi and J. Tsujii. (2004). Introduction to the Bio-Entity Recognition Task at JNLPBA. In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004)*, Geneva, Switzerland.

[12] J. Lafferty, A. McCallum and F. Pereira. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning*. Williamstown, MA, U.S.A.

[13] C. Lee, W. J. Hou and H.-H. Chen. (2004). Annotating Multiple Types of Biomedical Entities: A Single Word Classification Approach. In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004)*, Geneva, Switzerland.

[14] A. McCallum, Dayne Freitag and Fernando Pereira. Maximum Entropy Markov Models for Information Extraction and Segmentation. (2000). In *Proceedings of the International Conference on Machine Learning*. Williamstown, MA, U.S.A.

[15] A. McCallum and W. Li. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Conference on Natural Language Learning*, pages 188–191. Edmonton, Canada.

[16] K. M. Park, S. H. Kim, D. G. Lee and H. C. Rim. (2004). Boosting Lexical Knowledge for Biomedical Named Entity Recognition. In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004),* Geneva, Switzerland.

[17] L. R. Rabiner. (1989) A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, 77:257—286.

[18] C. J. van Rijsbergen. (1979). Information Retrieval. Butterworths, London.

[19] M. Rössler. (2004). Adapting a NER-System for German to the Biomedical Domain. In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004),* Geneva, Switzerland.

[20] B. Settles. (2004). Biomedical Named Entity Recognition Using Conditional Random Fields and Novel Feature Sets. In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004),* Geneva, Switzerland.

[21] F. Sha and F. (2003). Shallow Parsing with Conditional Random Fields. In *Proceedings of Human Language Technology-NAACL 2003,* Edmonton, Canada.

[22] E. F. Tjong, K. Sang and F. De Meulder. (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL-2003)*, pages 142-147. Edmonton, Canada.

[23] Y. Song, E. Kim, G. Geunbae Lee and B. K. Yi. (2004). POSBIOTM-NER in the shared task of BioNLP/NLPBA 2004. In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004),* Geneva, Switzerland.

[24] S. J. Zhao. (2004). Name Entity Recognition in Biomedical Text using a HMM model In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004),* Geneva, Switzerland.

[25] G. D. Zhou and J. Su. (2004). Exploring Deep Knowledge Resources in Biomedical Name Recognition. In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004),* Geneva, Switzerland.

ABOUT THE ORGANIZERS

Srinivasan Parthasarathy is an Assistant Professor (Associate Professor effective Fall 2005) in the Computer Science and Engineering Department at the Ohio State University. He received his MS ('96) and Ph.D ('00) degrees in computer science from the University of Rochester. His research interests include data mining, parallel and distributed systems, and bioinformatics. He is recipient of an Ameritech faculty fellowship in 2001, an NSF CAREER award, and a DOE early career principal investigator (ECPI) award, both in 2004. He has published over 90 articles in peer reviewed journals and conferences. His work in bioinformatics has recently garnered two best paper awards at the IEEE International Conference on Data Mining (2002) and the SIAM International Conference on Data Mining (2003).

Wei Wang is an Assistant Professor in the Computer Science Department at the University of North Carolina. She received her MS (‘95) in system science from the State University of New York at Binghamton and her Ph.D (‘99) in computer science from the University of California at Los Angeles. Her research interests include data mining, databases, and bioinformatics. She is the recipient of UNC Junior Faculty Development Award in 2003 and the recipient of an NSF CAREER award and Microsoft Faculty Fellowship, both in 2005. She has published more than 70 research papers in peer-reviewed journals and conferences.

Mohammed J. Zaki is an Associate Professor in the Computer Science Department at Rensselaer Polytechnic Institute. He received his M.S. ('95) and Ph.D. ('98) degrees in computer science from the University of Rochester. His research interests include the design of efficient, scalable, and parallel algorithms for various data mining techniques. He is especially interested in developing novel data mining techniques for applications such as bioinformatics and web mining. He received a CAREER Award from the National Science Foundation for his research in 2001 and an early career principal investigator award from the DOE in 2002. He has published over 100 articles in peer-reviewed journals and conferences.