ABSTRACT

ALGORITHMS IN COMPARATIVE GENOMICS

by
Satish Chikkagoudar

The field of comparative genomics is abundant with problems of interest to computer scientists. In this thesis, the author presents solutions to three contemporary problems: obtaining better alignments for phylogeny reconstruction, identifying related RNA sequences in genomes, and ranking Single Nucleotide Polymorphisms (SNPs) in genome-wide association studies (GWAS).

Sequence alignment is a basic and widely used task in bioinformatics. Its applications include identifying protein structure, RNAs and transcription factor binding sites in genomes, and phylogeny reconstruction. Phylogenetic descriptions depend not only on the employed reconstruction technique, but also on the underlying sequence alignment. The author has studied and established a simple prescription for obtaining a better phylogeny by improving the underlying alignments used in phylogeny reconstruction. This was achieved by improving upon Gotoh's iterative heuristic by iterating with maximum parsimony guide-trees. This approach has shown an improvement in accuracy over standard alignment programs.

A novel alignment algorithm named Probalign-RNAgenome that can identify non-coding RNAs in genomic sequences was also developed. Non-coding RNAs play a critical role in the cell such as gene regulation. It is thought that many such RNAs lie undiscovered in the genome. To date, alignment based approaches have shown to be more accurate than thermodynamic methods for identifying such non-coding RNAs. Probalign-RNAgenome employs a probabilistic consistency based approach for aligning a query RNA sequence to its homolog in a genomic sequence. Results show that this approach is more accurate on real data than the widely used BLAST and Smith-Waterman algorithms.

Within the realm of comparative genomics are also a large number of recently conducted GWAS. GWAS aim to identify regions in the genome that are associated with a given disease. The support vector machine (SVM) provides a discriminative alternative to the widely used chi-square statistic in GWAS. A novel hybrid strategy that combines the chi-square statistic with the SVM was developed and implemented. Its performance was studied on simulated data and the Wellcome Trust Case Control Consortium (WTCCC) studies. Results presented in this thesis show that the hybrid strategy ranks causal SNPs in simulated data significantly higher than the chi-square test and SVM alone. The results also show that the hybrid strategy ranks previously replicated SNPs and associated regions (where applicable) of type 1 diabetes, rheumatoid arthritis, and Crohn's disease higher than the chi-square, SVM, and SVM Recursive Feature Elimination (SVM-RFE).

ALGORITHMS IN COMPARATIVE GENOMICS

by
Satish Chikkagoudar

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Computer Science

Department of Computer Science

January 2010

APPROVAL PAGE

ALGORITHMS IN COMPARATIVE GENOMICS

Satish Chikkagoudar

Dr. Usman Roshan, Dissertation Advisor   Date
Assistant Professor of Computer Science, NJIT

Dr. Dennis Livesay, Committee Member   Date
Associate Professor of Bioinformatics and Genomics,
University of North Carolina, Charlotte

Dr. Zhi Wei, Committee Member   Date
Assistant Professor of Computer Science, NJIT

Dr. Barry Cohen, Committee Member   Date
Associate Dean, College of Computing Sciences, NJIT

Dr. Jason Wang, Committee Member   Date
Professor of Computer Science, NJIT

BIOGRAPHICAL SKETCH

Author: Satish Chikkagoudar

Degree: Doctor of Philosophy

Date: January 2010

Undergraduate and Graduate Education:

- Doctor of Philosophy in Computer Science,
  New Jersey Institute of Technology, Newark, NJ, 2010

- Master of Science in Computer Science,
  California State University, Chico, CA, 2004

- Bachelor of Engineering in Computer Science,
  Karnatak University, Dharwad, India, 2000

Major: Computer Science

Presentations and Publications:

Roshan U, Chikkagoudar S, Wei Z, Wang K, and Hakonarson H: A hybrid support vector machine strategy for ranking SNPs in genome-wide association studies, *Submitted.*

Roshan U, Wei Z, and Chikkagoudar S: Improved ranking of causal SNPs from genomewide association studies using the laplacian linear discriminant, *Accepted for poster presentation at ISMB/ECCB 2009, Stockholm, Sweden.*

Roshan U, Chikkagoudar S, Livesay DR: Searching for evolutionary distant RNA homologs within genomic sequences using partition function posterior probabilities. *BMC Bioinformatics* 2008, 9:61.

Chikkagoudar S, Roshan U, Livesay D: eProbalign: generation and manipulation of multiple sequence alignments using partition function posterior probabilities. *Nucleic Acids Res* 2007, 35(Web Server issue):W675-677.

Roshan U, Livesay DR, Chikkagoudar S: Improving progressive alignment for phylogeny reconstruction using parsimonious guide-trees. In: *Proceedings of the Sixth IEEE Symposium on BionInformatics and BioEngineering.* IEEE Computer Society; 2006: 159-164.

To My Loving Wife, Anne,

My Wonderful Parents, Dr. M. S. Chikkagoudar and Mrs. Ratna Chikkagoudar

And

My Siblings, Sunita and Gireesh

ACKNOWLEDGMENT

First and foremost, I thank Dr. Usman Roshan for being a helpful and resourceful advisor. I am extremely thankful to Dr. Dennis Livesay, Dr. Arthur Fridman, Dr. Narain Gehani, Dr. Barry Cohen, Dr. Zhi Wei, Dr. Jason Wang, Dr. Boris Verkhovsky, Uma Chandrashekar, Frank Bastry, Dr. Ansu Bagchi, Dr. Len Fisk, and Dr. Renee Renner for their help and guidance. Probalign web-server would not have been possible without the help of Gedaliah Wolosh and David Perel of NJIT. The CIPRES computing cluster, Dr. Mark Miller, and the CIPRES team at SDSC made my research possible and I am grateful to them. I thank Dongrong Wen, Suresh Solaimuthu, and Tao Wu for being extremely helpful friends and colleagues.

My family and friends supported me emotionally through the course of my education. My parents, Dr. M. S. Chikkagoudar and Mrs. Ratna Chikkagoudar, did a great job of encouraging me throughout my school years. They taught me to challenge myself and I am grateful to them. I thank my wife, Anne Lundquist, for being a patient wife and an excellent proofreader. Finally, I extend my special thanks to my other family members: Gireesh Chikkagoudar, Sunita Patil, Rohit Patil, Natalie Peretti, Mira Chikkagoudar, Dwight Lundquist, Mary Lundquist, Dr. H. B. Patil, Mahesh Patil, Nandeesh Patil, Margaret Blenkush, and Norman Lundquist.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

Chapter **Page**

TABLE OF CONTENTS
(Continued)

LIST OF TABLES

LIST OF TABLES
(Continued)

LIST OF FIGURES

Figure **Page**

LIST OF FIGURES
(Continued)

CHAPTER 1

INTRODUCTION

Since the human genome was decoded in 2001, advanced sequencing techniques such as high-throughput sequencing have resulted in the collection of large amounts of biomolecular data. Bioinformatics methods are required to analyze and make sense of such data. As a part of this dissertation, contemporary algorithms were developed for sequence alignment and phylogeny reconstruction. Algorithms to detect disease-associated/causal loci from single nucleotide polymorphism (SNP) genotype data were also developed in this research effort. Chapter 2 discusses a method for improved progressive alignment for phylogeny reconstruction using parsimonious guide-trees. Chapter 3 discusses an alignment based RNA homology search algorithm that uses partition function posterior probabilities. The web-servers developed to allow an Internet based access to Probalign and Probalign-RNAgenome are described in Chapter 4. Chapter 5 describes discriminative machine learning techniques that can be used to detect disease-associated/causal loci from SNP genotype data.

Sequence alignment and phylogeny reconstruction are widely used techniques in bioinformatics. Sequence alignments are used as inputs for phylogeny reconstruction programs. The sections below will introduce basic concepts of sequence alignment, phylogeny reconstruction and genome-wide association studies.

1.1 Sequence Alignment

Research suggests that evolutionarily conserved regions and patterns in sequences are biologically significant. The motivation behind alignment of biological sequences is to identify such regions or patterns in sequences [1].

In layman's terms sequence alignment is nothing but "inexact" string matching. Indels or gaps are inserted in those positions of the resulting alignment that cannot be properly matched. According to Jones et al. [1], a multiple sequence alignment A of k input sequences $S_1, S_2, \ldots, S_k$ is k strings $S_1', S_2', \ldots, S_k'$ such that each of the resulting strings $S_1', S_2', \ldots, S_k'$ are of equal length and are an extension of the corresponding/ respective sequence with the inclusion of spaces/gaps.

```
>sequence1
 AACTUU
>sequence2
 -ACT--
```

Figure 1.1 Example of an alignment.

Alignments between several sequences are called multiple sequence alignments (MSA). Such alignments help identify conserved regions/patterns within a sequence. Some conserved regions/patterns that cannot be easily discerned in pairwise alignments can be easily identified using multiple sequence alignments. Multiple sequence alignments can also be used for phylogeny reconstruction, protein functional site detection, protein structure prediction, and RNA structure prediction [2].

Many alignment algorithms utilize an affine gap penalty scheme for aligning sequences. Gaps are penalized using gap open and gap extension penalties in order to discourage excessive gaps in alignments. The dynamic programming formulae for a sequence alignment are [2, 3]:

$$G_{A_{i,j}} = \max\begin{cases} G_{A_{i-1,j}} + ext \\ S_{i-1,j} + g + ext \end{cases} \qquad \boxed{\begin{array}{l} initially: \\ G_{A_{0,j}} = -\infty \\ G_{B_{i,0}} = -\infty \\ S_{0,j} = g + j * ext \\ S_{i,0} = g + i * ext \\ S_{0,0} = 0 \end{array}}$$

$$G_{B_{i,j}} = \max\begin{cases} G_{B_{i,j-1}} + ext \\ S_{i,j-1} + g + ext \end{cases}$$

$$S_{i,j} = \max\begin{cases} S_{i-1,j-1} + \delta(i-1, j-1) \\ G_{A_{i,j}} \\ G_{B_{i,j}} \end{cases} \tag{1.1}$$

Where, given that one is aligning two sequences $sequence1[1.n]$ and $sequence2[1.m]$ at positions i and j respectively [2,3]:

G_A=matrix that handles/stores the best alignment that ends with a gap in $sequence2$

G_B=matrix that handles/stores the best alignment that ends with a gap in $sequence1$

S=matrix that handles/stores the case in which $sequence1$ and $sequence2$ are aligned (with a match/mismatch)

δ=score of aligning the residues/bases at $sequence1_i$ and $sequence2_j$

g=gap open penalty

ext=gap extension penalty

Multiple sequence alignments are usually scored using a Sum of Pairs (SP) scoring scheme. Several scoring matrices are available for scoring substitutions in alignments. PAM [1], BLOSUM [1], and VTML [4] are examples of commonly used scoring matrices. The SP score of a multiple sequence alignment is the sum of the SP scores of its constituent pairwise alignments [1, 2].

```
Gap open penalty = -4
Gap extend penalty = -1
Scoring matrix: matches (AA,CC,GG,TT,UU) = 1

>sequence1
AACTUU
>sequence2
-ACT--

SP score of the above alignment = -4+1+1+1-4-1 = -6
```

Figure 1.2 Scoring an alignment.

Benchmark sequence databases and scoring programs such as QScore can be used to compare and benchmark the performance of multiple sequence alignment programs. BAliBASE [5] and SABmark [6] are examples of popular protein sequence benchmark databases.

Probalign

Several multiple sequence alignment software packages are open-source and are readily available over the Internet. ClustalW [7], MUSCLE [4], MAFFT [8], Probcons [9], and Probalign [10] are examples of multiple sequence alignment programs. MAFFT [8], Probcons [9], and Probalign [10] are recent alignment strategies that are among recent programs with the highest accuracies on BAliBASE [5] and other common benchmarks (i.e., HOMSTRAD [32] and OXBENCH [33]).

Both Probcons and Probalign compute maximal expected accuracy alignments using posterior probabilities. In Probcons, posterior probabilities are derived using a Hidden Markov Model (HMM) whose parameters have been estimated via supervised learning on BAliBASE unaligned sequences. Probalign, which is largely based on the Probcons scheme, derives the posterior probabilities from input data by implicitly examining suboptimal sum-of-pair alignments using the partition function methodology for alignments [10]. Probalign alignments have been shown to have a statistically significant improvement over Probcons, MAFFT and MUSCLE on all three alignment benchmarks introduced above [10].

Probalign uses partition function matrices to generate posterior probabilities. The dynamic programming formulae that are used to generate partition function matrices in Probalign are [10]:

$$\begin{aligned} Z^M_{i,j} &= (Z^M_{i-1,j-1} + Z^E_{i-1,j-1} + Z^F_{i-1,j-1})e^{s(x_i,y_i)/T} \\ Z^E_{i,j} &= Z^M_{i,j-1}e^{g/T} + Z^E_{i,j-1}e^{ext/T} \\ Z^F_{i,j} &= Z^M_{i-1,j}e^{g/T} + Z^F_{i-1,j}e^{ext/T} \\ Z_{i,j} &= Z^M_{i,j} + Z^E_{i,j} + Z^F_{i,j} \end{aligned} \tag{1.2}$$

Here, $s(x_i,y_j)$ represents the score of aligning residue x_i with y_j, g is the gap open penalty, and *ext* is the gap extension penalty. T is the thermodynamic temperature and it is used to define the extent to which suboptimal alignments are considered. The matrix $Z^M_{i,j}$ represents the partition function of all alignments ending in x_i paired with y_j. Similarly, $Z^E_{i,j}$ represents the partition function of all alignments in which y_j is aligned to a gap and $Z^F_{i,j}$ all alignments in which x_i is aligned to a gap.

Posterior probability is then calculated using the above-mentioned partition function matrices and the following formula [10]:

$$P(x_i \sim y_i) = \frac{Z^M_{i-1,j-1}Z^M_{i+1,j+1}}{Z}e^{s(x_i,y_i)/T} \tag{1.3}$$

The posterior probability matrix $P(x_i \sim y_i)$ is used to compute a maximal expected accuracy alignment A using the following recursive formula [10]:

$$A(i,j) = \max \begin{Bmatrix} A(i-1, j-1) + P(x_i \sim y_i) \\ A(i-1, j) \\ A(i, j-1) \end{Bmatrix} \qquad (1.4)$$

Web interfaces are available for the multiple sequence alignment programs discussed above. eProbalign is the web server version of Probalign and provides a convenient platform to visualize alignments, generate images, and manipulate the output by average column posterior probabilities. The average column posterior probability that is computed by eProbalign can be considered as a measure of column reliability where columns with higher scores are more likely to be correctly aligned and biologically informative.

1.2 Phylogeny Reconstruction

Phylogenies are a fundamental tool for understanding the evolutionary history of species [1]. As mentioned earlier, multiple sequence alignments can show evolutionarily conserved regions in sequences. This feature of multiple sequence alignments can be exploited to reconstruct the evolutionary history of species. Therefore, the most important input to a phylogeny reconstruction method is a multiple sequence alignment.

Phylogenies are represented as trees. Nodes of a phylogenetic tree represent species, while the edges represent genetic/evolutionary distance between species. Phylogenetic trees can be either rooted or unrooted.

The two main approaches for phylogeny reconstruction are Maximum Parsimony (MP) and Maximum Likelihood (ML). Both approaches are known to be NP-hard. However, in practice, heuristic ML implementations are orders of magnitude slower than heuristic MP implementations [14].

The objective of the MP approach is to reconstruct a tree by minimizing mutations [1]. Standard heuristics for solving MP are hill-climbing strategies that use the Tree Bisection and Reconnection (TBR) technique for performing local moves [14].

The absence of "true" phylogenetic or evolutionary relationship data makes it difficult to evaluate the quality of a reconstructed phylogenetic tree. Hence, phylogenetic reconstruction methods are evaluated using simulation. Given the true tree (which is known since the data is simulated) and an estimated tree, the Robinson-Foulds distance [23] can be used to measure accuracy. This is a standard measure of evaluating tree accuracy in phylogenetics and measures the number of false positive and false negative clades in the estimated tree. The error rate is presented as percentages (between 0 and 100).

Benchmark databases of phylogenies are used to test phylogeny reconstruction programs and strategies. The quality of a reconstructed phylogeny is measured by comparing it to a "true" phylogeny that has been reconstructed by experts and is a part of a benchmark database.

1.3 Genome-wide Association Studies

Genome-wide association studies conducted to date have identified SNPs associated with several diseases as well as various phenotypes and drug responses [71]. Such SNPs can be found in growing online databases such as SNPedia (http://www.SNPedia.com). The study on seven common diseases conducted by the Wellcome Trust Consortium is one of the largest to date [74]. It reported several significant SNPs from 2,000 case subjects per disease and 3,000 shared controls.

The standard method of detecting disease associated SNPs from a genome wide association study is to perform a χ^2 (chi-square) test on each SNP and select the k top ranked ones (or those below a p-value threshold) for further study [64, 68, 73, 74, 75]. Once the significant SNPs are identified, one can use them to predict the disease risk of healthy individuals as well as to identify genes and their regions of interest.

χ^2 (Chi-square) Test

The independence of variables or populations can be tested using a χ^2 test. The χ^2 test statistic can be written as the following formula:

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i} \tag{1.5}$$

Where, n = number of observations or outcomes

O_i = observed frequency of the i^{th} observation or outcome

E_i = expected frequency of the i^{th} observation or outcome

The null hypothesis used in the χ^2 test is that the variables are independent. The p-value for a given χ^2 statistic value can be obtained by referring to a χ^2 distribution table.

CHAPTER 2

IMPROVING PROGRESSIVE ALIGNMENT FOR PHYLOGENY RECONSTRUCTION USING PARSIMONIOUS GUIDE-TREES

2.1 Introduction

Phylogenies are a fundamental tool for understanding the evolutionary history of species [1]. The most important input to a phylogeny reconstruction method is a multiple sequence alignment. The progressive alignment strategy of Feng and Dolittle [12] is a fast and widely used heuristic for aligning multiple sequences to a guide-tree (i.e., phylogenetic tree sequence alignment). For example, the popular ClustalW program [13] uses a progressive alignment combined with improvements built around it. Guide trees for progressive alignment are usually obtained by simple distance-based approaches such as neighbor joining or UPGMA [14], where distance matrices are constructed using pairwise alignments.

Most previous phylogenetic reconstruction studies have focused on constructing optimal trees with the alignment fixed. However, the input alignment is known to affect the reconstructed phylogeny [15, 16, 17]. Consequently, improving the alignment input could lead to better phylogenies. A simple MP iterative refinement method that is based on Gotoh's [18] doubly nested randomized iterative technique can result in significantly improved sequence alignments for phylogeny reconstruction. This research effort compares this approach to the standard ClustalW, and different stages of MUSCLE on simulated data.

2.2 Methods

2.2.1 Phylogeny Reconstruction and Alignment

Maximum parsimony (MP) and maximum likelihood (ML) are two widely used optimization criteria for phylogeny reconstruction [14]. Both are known to be NP-hard; however, in practice, heuristic ML implementations are orders of magnitude slower than MP [14]. Consequently, this dissertation only examines MP for constructing phylogenies in this preliminary investigation; investigation of ML is left to a later study. Standard heuristics for solving MP are hill-climbing strategies which use the Tree Bisection and Reconnection (TBR) technique for performing local moves [14]. These can be found in software packages like PAUP* [19].

Like MP and ML phylogenetic reconstruction, standard optimization criteria for multiple sequence alignment, i.e., sum-of-pairs and phylogenetic tree alignment [1] are also NP-hard. Sum-of-pairs (SP) aims to maximize the sum of pairwise similarity between the input sequences. Phylogenetic tree alignment, on the other hand, aims to minimize dissimilarity along the edges of a given tree. The progressive alignment strategy [12] has been adapted into most software packages for alignment, the most popular being ClustalW [13] because of its speed and accuracy.

Various programs have implemented improvements around the basic progressive alignment. ClustalW implements ideas such as sequence weighting and automatic gap penalties that are designed to improve the alignment based on biologically sound assumptions [13]. ClustalW uses neighbor joining for a guide-tree. MUSCLE [4] is a three-stage program each of which are studied separately for this report. Stage I is the basic progressive alignment on a UPGMA guide-tree. Stage II is Gotoh's iterative heuristic [18] but without SP optimization, i.e., compute alignment on a UPGMA tree, compute UPGMA tree on alignment, recompute alignment on UPGMA tree, and iterate until the UPGMA tree does not change. Stage III is a SP optimization on the alignment from stage II.

2.2.2 Simulation

Simulations are commonly used to evaluate phylogenetic accuracy since there is no way of knowing "true" evolutionary trees [20]. The ROSE software package [21] implements the HKY85 [22] model of DNA sequence evolution, but also allows for insertions and deletions. Given the true tree (which is known since the data is simulated) and an estimated tree, the Robinson-Foulds distance [23] can be used to measure accuracy. This is a standard measure of evaluating tree accuracy in phylogenetics and measures the number of false positive and false negative clades in the estimated tree. The error rate is presented as percentages (between 0 and 100).

2.3 Improved Progressive Alignment

Gotoh [18] introduced a doubly nested randomized iterative method that iterated between progressive alignments and distance-based UPGMA phylogenies. In this work, Gotoh's approach is modified by alternating between MP trees and progressive alignments. The output is the pair of alignment and tree with the best MP score. This heuristic is implemented using the MUSCLE program (for computing the progressive alignment) and PAUP* (for computing MP trees) and is called MUSCLE-PARS (see Figure 2.1). MUSCLE-PARS is specifically designed to find alignments and phylogenies that optimize the MP score, and thus is likely to be more appropriate for phylogeny-centric applications, i.e., predicting functional sites with phylogenetic motifs [24]. MUSCLE-PARS strictly follows the order of the tree in aligning sequences. PAUP* implements various hill-climbing heuristics for solving MP. The MP heuristic that is used by

MUSCLE-PARS builds a starting tree by adding sequences in the order of their closeness (see [14] for more details). Once the tree is constructed, a TBR-based standard hill-climbing search is applied to it. The initial starting tree for the search can also be built by adding sequences in a random order instead of their closeness; this produces a randomized search heuristic since each time the search starts from a different tree. The former deterministic search for MP is used so that MUSCLE-PARS is also deterministic. A thorough study of the randomized version of MUSCLE-PARS is left to a later study.

```
Input: unaligned sequences, initial guide-tree $T$, number of iterations $n$
Output: alignment $A^*$ and guide-tree $T^*$
Algorithm:
    (1) Set best score $bs$ to infinity.
    (2) Compute MUSCLE progressive alignment $A$ on guide-tree $T$
    (3) Compute MP score $MP(T,A)$ of tree $T$ on alignment $A$.
    (4) If $MP(T,A) < bs$ then
            set $bs = MP(T,A)$, $A^* = A$, and $T^* = T$
    (5) Compute MP tree $T$ on $A$ using PAUP*.
    (6) If number of iterations not done then
            go to 2
        else
            return $A^*$ and $T^*$.
```

Figure 2.1 Description of MUSCLE-PARS.

MUSCLE-PARS differs from Gotoh's original implementation in several key ways. First, the original method of Gotoh [18] used UPGMA trees instead of MP. Second, Gotoh's method performed SP optimization on the progressive alignment before recomputing a phylogeny on it. MUSCLE-PARS does not perform this additional optimization step because it does not necessarily improve accuracy and extends running time (data not shown here). Third, the stopping criterion for Gotoh's method is when the UPGMA tree does not change; Gotoh's method usually reaches convergence in a few iterations. MUSCLE-PARS uses parsimony trees (that may be deterministic or randomized), which provides no guarantee of convergence; alignments and trees could get worse or improve with iterations. If the same alignment is obtained in two consecutive iterations, the MP trees (which are used for constructing the alignment of the following iteration) may not be the same if randomized heuristics are used. And fourth, the alignment outputted from Gotoh [18] is the one from the most recent iteration. MUSCLE-PARS outputs the alignment and tree with the best MP score over all the iterations.

2.4 Experimental Design

This dissertation work compares ClustalW, and MUSCLE in its three different stages to two variants of MUSCLE-PARS using default scoring matrices and gap penalties. The scoring matrices and gap penalties of the MUSCLE variants and MUSCLE-PARS are exactly the same; the only difference is in the guide-tree iterations. The abbreviation MUSCLE-PROG refers to stage I of MUSCLE, MUSCLE-UPGMA refers to stage II, and MUSCLE refers to the final stage III alignment. Additionally, this work presents two variants of MUSCLE-PARS. In the first, which is called MUSCLE-PARS, the initial guide-tree is the UPGMA one constructed on pairwise alignment distances. In the second one, which is called MUSCLE-PARS2, the initial guide-tree is the one used in the last iteration of MUSCLE-UPGMA. MP phylogenies are constructed on all the alignments (on each simulated dataset) using a more thorough TBR search heuristic than the basic one used in MUSCLE-PARS (available upon request). Since PAUP* was used in MUSCLE-PARS, it is used for constructing MP phylogenies on all alignments. Simulation parameters are selected such that the MP tree on the true alignment has, at most, 15% error. Birth-death model trees produced using the r8s software package [25] are used. Birth-death trees produced by r8s are scaled to be ultrametric by default, which means that the evolutionary distance from the root to each leaf is the same. Biological trees on real data are not necessarily ultrametric; therefore, to deviate the tree from ultrametricity each edge length is randomly multiplied by a deviation factor as described in [26]. A deviation of 1 means no deviation, 2 means small, and 4 is moderate deviation. The edge lengths of each tree are multiplied by scaling factors of 16, 32, and 64 to produce different levels of evolutionary rates. For each setting of deviation and scale, this effort generated 20 model trees of sizes 100, 200, and 400 taxa. Thus, producing a total of 360 different model trees.

For each model tree, DNA sequences are generated using ROSE under the HKY85 [22] model with transition/transversion ratio set to 2. This research effort studies two sequence lengths used at the root, 500 and 1000, and examines two different indel probabilities of 0.00005 and 0.0005 (see [21] for more details). On each of the 360 model trees, DNA sequences were evolved for each setting of sequence length and indel probability; thus, producing a total of 1,440 simulated datasets.

2.5 Experimental Results

For each set of simulated unaligned sequences, ClustalW, MUSCLE (all three stages), and MUSCLE-PARS (both variants) alignments are computed. Subsequently, MP trees are constructed using a thorough TBR search heuristic. The accuracy of each phylogeny, computed using the RF distance, is compared against the true tree. The average error rate for each parametric setting is reported in tables A.1 and A.2. The improvement, in terms of percentage differences, is also provided for the best scoring alignment. The improvement in MUSCLE-PARS1 and MUSCLE-PARS2 error rates over the best error rate of the other methods are also reported. While the average gain is modest, the overall results clearly indicate that improvement when using the two MUSCLE-PARS methods is a robust result. The results follow some of the general trends one would expect to see in simulation studies. For example, the error rates decrease as the sequence length increases. Conversely, error rates tend to increase as the evolutionary rates, number of taxa, deviations, or indel probabilities increases, all of which are known to make the phylogeny estimation problem harder. However, trees at evolutionary rates of 32 fare better than 16. Overall MUSCLE-PARS1 and MUSCLE-PARS2 have the lowest error rates. At sequence lengths of 1000 and low indel probability of 0.00005 the improvement using MUSCLE-PARS is the smallest (especially at 100 taxa), if any at all. A closer look is taken at the part of the parameter space where the improvement is over 1% in topological accuracy.

There are nine parametric settings at which MUSCLE-PARS has an error rate lower than 1% than the other methods. Out of those seven are for sequence lengths of 500. Thus, MUSCLE-PARS can be most effective when sequence lengths are short relative to the number of sequences. On six of these settings the indel probability is 0.0005 (the higher value) thus showing that MUSCLE-PARS can be useful for data that has undergone a modest number of insertions and deletions. The largest improvement is of 2.2% for 200 sequences, 500 sequence length, 64 scaling, 4 deviation, and 0.0005 indel probability, which can be considered a hard setting. A curious observation is that MUSCLE has high error rates, especially when considering high evolutionary rates and indel probabilities. In fact, the error rates sometimes go above 25%, which is much higher than that of the other methods. Recall that MUSCLE computes a SP optimization in stage III after the progressive alignments are done. It is conjectured that this significantly decreases the quality of the alignment for phylogeny reconstruction. However, for other tasks, such as aligning structurally conserved regions, it may be more appropriate as seen from performance on BAliBASE [5] structural alignment benchmarks. When considering protein data, this anti-correlation between phylogeny reliability (using bootstraps) and BAliBASE accuracy was also noticed. These observations underscore the reality that no single assessment strategy can be considered perfect when evaluating alignments and phylogenies.

2.6 Conclusions

The above-mentioned experiments on data show that MUSCLE-PARS1 and MUSCLE-PARS2 produce phylogenies of better accuracy than those on ClustalW, MUSCLE-PROG, MUSCLE-UPGMA, and MUSCLE. Furthermore, MUSCLE-PARS is efficient in the running time required to produce an alignment and phylogeny (data not shown here), which means it can be used to analyze datasets containing even hundreds to thousands of sequences. MUSCLE-PARS can be expected to quickly produce very good starting trees for expensive simultaneous alignment and phylogeny reconstruction local search strategies, such as those conducted in Poy [27] and statistical alignment packages [28].

MUSCLE-PARS can easily be implemented using existing available software packages with a simple Perl script.

CHAPTER 3

RNA HOMOLOGY SEARCH USING PARTITION FUNCTION POSTERIOR PROBABILITIES

3.1 Introduction

The importance of RNA within cellular machinery and regulation is well established [35, 36]. Consequently, a proper understanding of RNA structure and function is vital to a more complete understanding of cellular processes. It is conjectured that the human genome contains several thousand yet undiscovered ncRNAs that play critical roles throughout the cell. Profile-sequence and structure-sequence methods, such as HMMER [37] and INFERNAL [38], are commonly used to identify RNA homologs within much larger genomic segments. However, the requirement of a reliable family alignment and/or structure diminishes the utility of these approaches. This can happen especially when searching for evolutionary distant homologs or the query RNA sequence is surrounded by unalignable flanking nucleotides. In fact, homologous sequences below 60% pairwise identity are generally too difficult for current methods [39]. Simple pairwise alignment approaches are commonly used when sufficient familial data is not available. The SSEARCH program [40], a popular implementation of the Smith-Waterman algorithm, is frequently used for finding RNA homologs in genomic sequences. Moreover, it is a commonly used benchmark that new homology search methods are compared against [41-44]. The NCBI BLAST program [45], which is also a local alignment algorithm, is faster than SSEARCH but much less sensitive.

SSEARCH and BLAST both search for optimal local alignments, with BLAST sacrificing sensitivity for speed. Conversely, the maximal expected accuracy approach is based on suboptimal alignments. Here, sequences are aligned using posterior/match probabilities within pairwise alignments. These probabilities can be computed using partition function dynamic programming matrices, introduced by Miyazawa [46] and later studied by others [47, 10], or pairwise HMMs as done in ProbconsRNA [9]. Partition function posterior probabilities are analogous to nucleotide-nucleotide frequency counts estimated from an ensemble of suboptimal alignments (see ref. [14] for more details). The recently implemented partition function approach within the program Probalign [10] outperforms other leading multiple aligners (Probcons [9], MAFFT [8], and MUSCLE [4, 34]) on three different protein alignment benchmarks (BAliBASE [31], HOMSTRAD [32], and OXBENCH [33]).

While Probalign was designed for global alignment, its performance on datasets of heterogeneous length [10] suggests an affinity for local alignment. In this work, a slightly modified Probalign version attuned to local alignment search is implemented. Its performance is studied on the pairwise RNA-genome homology search problem for divergent sequences and when the query is flanked by genomic nucleotides. The above-mentioned implementation of Probalign is compared to SSEARCH, BLAST, ClustalW [13], and HMMER (with single sequence profiles). ClustalW (with zero end gaps) is included in this study due its wide usage in solving different alignment problems. In addition, ClustalW serves as an analogous example of a global multiple alignment method applied to this problem. In this work, a benchmark of divergent RNA-genomic alignments using real DNA and RNA sequences was constructed from the EMBL [48] and RFAM [49] databases, respectively. In order to maintain a reasonable level of difficulty and tractability for the experiments, each genomic sequence in this benchmark is at least 5K and at most 16K nucleotides in length. For added difficulty and to simulate practical conditions where exact 5' and 3' ends of ncRNA are unknown, real genomic flanks of size 50, 100, and 150 nucleotides are added to the query RNA of each dataset.

INFERNAL was specifically omitted from this investigation for several reasons. First, and most importantly, (as discussed above) the utility of profile-sequence and structure-sequence alignment methods is limited by experimental data. At large evolutionary distances and with unalignable genomic flanks surrounding the query, which is the particular focus of this study, obtaining reliable RNA family alignments is considerably difficult. Second, the cmsearch program of the INFERNAL suite is, in part, used for constructing RFAM families from which the benchmark is constructed. Additional sequences found using INFERNAL were added to the RFAM seed alignments [50]. Finally, cmsearch is used in intermediary steps of producing the benchmark (explained in the Methods Section below). In light of all these facts, it would be inappropriate to include INFERNAL in the experiments. HMMER is included in the experiments using both global-local and local-local alignment models (i.e., -g and -f, -s options); however, the HMMER model is constructed using single sequence queries (without flanks) from the benchmark. In this way, there is a reasonable comparison to the other sequence-based programs in the test set. In the remainder of the report this setting of the program is referred to as just HMMER.

In this study, Probalign is found to have overall highest accuracies on the full benchmark. It leads by 10% accuracy over SSEARCH (the next best method) on 5 out of 22 families. On datasets restricted to maximum of 30% sequence identity, Probalign's overall median error is 71.2% vs. 83.4% for SSEARCH (the next best method). This difference has Friedman rank test P-value less than 0.05. Furthermore, on these datasets, Probalign leads SSEARCH by at least 10% on five families whereas SSEARCH leads Probalign by the same margin on two families out of a total of fourteen. This report also demonstrates that the Probalign mean posterior probability, compared to the normalized SSEARCH Z-score, is a better discriminator of alignment quality. The Probalign mean posterior probability has Receiver Operator Characteristic (ROC) area under curve of 0.834 compared to 0.806 of the normalized SSEARCH Z-score.

Note that the performance of RNA homology search programs was examined previously by Freyhult et al. [41]. Their benchmark and goals, however, were considerably different than those of this study. They studied RNA homology searches within RFAM RNA sequence databases without genomic flanks, and considered only a single genomic search example. This endeavor is specifically interested in the performance of programs for finding low sequence similarity RNA homologs (with flanks) in long genomic sequences.

3.2 Results

First, the mean error was computed for each method within each RNA family by averaging over all pairwise alignment scores belonging to that family. Then, the overall error of each method was computed as the average score across all families.

Full Benchmark with Query Flanks

The full benchmark containing query RNAs with flanks constitute 13,716 datasets. HMMER is excluded when unalignable flanks are present since these will only confound the model. Table 3.1 lists the overall mean and median error of all methods on the full benchmark. Probalign's improvement is statistically significant lowest on datasets restricted to max 30% sequence identity. On these datasets it leads SSEARCH (the next best method) by 6.5% in mean error and 11.2% in median error.

Table 3.1 Mean and Median Percent Error for All Methods on the Full Benchmark (13,716 Datasets) Including Query RNAs with Flanks of Size 50, 100, and 150

Mean and median error	Probalign	SSEARCH	BLAST	ClustalW
Complete benchmark	**35.3 | 30.7**	38.7 | 33.2	41.0 | 34.0	47.6 | 50.3
Datasets with pairwise sequence identity at most 30%	**66.5*| 71.2***	73.0 | 83.4	75.9 | 85.3	82.9 | 85.0

Note: BLAST does not return an alignment in 425 datasets and hence they are omitted from the calculations. HMMER is not shown since queries with unalignable flanks cannot be used to produce a reliable model. There are 14 families that contain datasets with at most 30% sequence identity. Probalign has overall lowest mean and median error. Bold indicates the best performance; the difference is larger on datasets with low sequence identity and significant with P-value < 0.05 (indicated by *).

Table 3.2 lists the error rates of Probalign and the next best method, SSEARCH, on each RFAM family. Probalign leads by 10% on a total of five families, namely T-box, Intron group I, signal recognition particle (eukaryotic), transfer RNA, and elenocysteine insertion sequence. The maximum improvement by SSEARCH over Probalign is on the U4 spliceosomal RNA family by 3.1%. Column two of the table lists the Probalign and SSEARCH error on datasets restricted to maximum 30% sequence identity. There are fourteen families containing datasets that satisfy this criterion. Out of the total fourteen, Probalign leads by at least 10% on five families whereas SSEARCH leads Probalign by at least same margin on two families.

Table 3.3 looks at the effect of increasing query flank size on the accuracy of all methods. As expected, all methods yield higher error as the query RNA flank size increases. However, Probalign still has the statistically significantly lowest error (P-value < 0.05).

Table 3.2 Mean Probalign and SSEARCH Percent Error Shown for Each RFAM Family in the Full Benchmark and for Datasets with Maximum Pairwise Sequence Identity of 30%

RFAM Family	Complete benchmark dataset			Subset with pairwise identity up to 30%		
	Probalign	SSEARCH	Difference	Probalign	SSEARCH	Difference
5S_rRNA	22.7	20.7	-2.0		Zero datasets	
U1 (4)	15.0	15.6	0.6	87.3	100.0	12.7
tRNA (256)	62.0	74.4	12.3	69.8	84.8	15.0
RNaseP_bact_a	34.0	33.0	-1.0		Zero datasets	
RNaseP_bact_b	29.0	29.1	-0.1			
U3	41.3	38.8	-2.5			
U4 (8)	25.3	22.2	-3.1	52.8	11	-41.8
SRP_euk_arch (132)	43.8	56.4	12.6	62.1	78.0	15.9
tmRNA (180)	32.0	36.3	4.3	50.5	59.8	9.4
Intron_gpI (4)	67.4	80.1	12.7	100.0	100.0	0.0
SECIS (208)	82.3	93.9	11.5	87.9	100.0	12.1
IRE (216)	44.4	48.7	4.2	88.7	96.5	7.7
THI	29.5	30.1	0.6		Zero datasets	
Hammerhead_1	43.7	46.0	2.3			
Purine (4)	16.2	16.4	0.2	17.4	1.8	-15.6
Lysine (16)	48.0	57.3	9.3	73.1	100.0	26.9
SRP_bact (80)	28.5	25.7	-2.8	62.6	65.0	2.3
SSU_rRNA_5 (4)	30.5	32.4	1.9	39	61	22
T-box	27.4	46.0	18.6		Zero datasets	
glmS (4)	23.4	21.0	-2.4	73.8	78.4	4.6
RNaseP_arch (8)	32.4	34.0	1.6	87	100.0	13
IRES_Cripavirus	5.7	3.9	-1.8		Zero datasets	

Note: Unlike Table 3.1 above, where some datasets are omitted due to BLAST, all datasets of the benchmark are considered here. Difference is always calculated as the SSEARCH error minus Probalign error, meaning positive numbers indicates Probalign outperforms SSEARCH. Shown in parenthesis is the number of datasets in each family with maximum pairwise sequence identity of 30% (the same query RNA but with different flank sizes is considered a separate dataset).

Table 3.3 Mean Percent Error as a Function of Query RNA Flank Size

Query RNA flank size	Probalign	SSEARCH	BLAST	ClustalW
50	**35.4***	39.3	41.9	48.5
100	**36.8***	40.8	44.5	51.4
150	**38.5***	43.3	45.9	53.2

Note: For each flank size there are 3,429 datasets (see Methods Section for description of benchmark). As in Table 3.1 about 105 datasets per flank size are omitted on which BLAST does not return any output. Bold indicates the best performance and * indicates Friedman rank test P-value < 0.05.

Benchmark Without Query Flanks

In order to compare the programs against HMMER, those datasets with no query RNA flanks (a total of 3,429) are separated from the benchmark. Each of these query RNAs can be used to specify a model in HMMER since misleading flanks are now absent. From Table 3.4 it can be seen that HMMER does not perform very well with single sequence profiles, which is not surprising as using it in this way (single sequence vs. multiple sequence profiles) clearly goes against its intended usage. Probalign has the lowest mean and median error on datasets restricted to maximum pairwise identity of 30%, leading by at least 18% over SSEARCH, the next best method.

Table 3.4 Mean and Median Percent Error for All Methods on the Benchmark without Query RNA Flanks (3,429 Datasets)

Mean and median error	Probalign	SSEARCH	BLAST	ClustalW	HMMER
Complete benchmark	**30.8** | 30.4	31.4 | 22.1	32.0 | **20.9**	37.9 | 38.5	44.9 | 44.7
Datasets with pairwise sequence identity atmost 30% (14)	**62.4** | **59.5**	70.8 | 94.5	78.4 | 100.0	74.5 | 97.5	96.7 | 100.0

Note: Probalign has lowest mean and median error on low sequence identity datasets.

Discriminating True from False Alignments

In order for the evaluated methods to be of practical utility in ncRNA searches, alignments found when there is no target-query match (a common real-world scenario), should be of poorer quality than the alignments above where target-query matches were always present. A false dataset of query-target pairs where the query and target were randomly selected from distinct RFAM families (see the Alignment Quality Measures Sub-Section under the Methods Section) was generated in order to assess the discriminative ability of Probalign and SSEARCH (the two best scoring methods above). The size of the false dataset is 13,716, exactly the same as the real dataset used above. Concatenating the real and false datasets results in 27,432 target-query pairs that were subsequently aligned using both methods. An alignment on a false positive dataset or an alignment with 100% error on the benchmark is classified as a false positive. An alignment on the benchmark with less than 100% error is classified as a true positive. A good discriminator would have a high value on alignments with high accuracy and low value on alignments with 100% error on benchmark datasets or on the false positive dataset. In this case, this research endeavor is interested in the quality of the Probalign mean column posterior probability and the SSEARCH normalized Z-score as alignment discriminators.

In order to evaluate a discriminator, an *adhoc* threshold needs to set. For example, one may choose to classify all alignments above 0.5 Probalign mean column posterior probability to be correct hits and incorrect otherwise. In order to eliminate the arbitrariness of such a definition, Receiver Operating Characteristic (ROC) analysis is employed. Along the ROC curve, true and false positive prediction values are plotted for a series of less stringent thresholds. The further the ROC curve is to the left, the better the method is; the diagonal indicates a method based on random guesses. As can be clearly seen in Figure 3.1, both methods perform significantly better than random. However, the analysis also clearly indicates that Probalign is better able to discriminate true from false target-query pairs. Probalign has an area under curve of 0.834 whereas SSEARCH has one of 0.806. The improved performance of Probalign is most striking at false positive rates between 2 and 40%.

Computational Running Time and Memory Requirements

The current Probalign implementation is not as sophisticated as its SSEARCH counterpart, and therefore is much slower in comparison to the SSEARCH time. However, in practice it never takes more than a few seconds on any of the datasets. The average Probalign running time on the benchmark is 5.4 seconds compared to 0.04 seconds of SSEARCH, 0.5 seconds of ClustalW, 0.003 seconds of BLAST, and 0.14 seconds of HMMER (hmmsearch). These running times were computed on 2.4 GHz AMD Opteron 64 bit machines.

3.3 Discussion

A standard technique for discovering new RNAs, in the absence of queries, is to align genomic fragments and search the alignment for significant structural conservation. QRNA [51] RNAz [52] and MSARI [53] are some well-known programs frequently used for this purpose. Their performance, of course depends upon the underlying sequence alignments. This work suggests that Probalign genomic alignments may align hidden (but related) RNA better than standard methods when given two genomic sequences. As a result it could produce more informative alignments for RNA detection programs such as the ones listed above.

Several improvements to Probalign are currently underway. A full Probalign-local implementation would include a Smith-Waterman implementation of posterior probability local alignment, as done in the Proda [54] program. It is expected that such an implementation will produce better mean posterior probabilities estimates of the alignment quality since it would exclude unrelated genomic flanks.

In big-O notation, Probalign's worst-case running time and memory requirements for pairwise alignment is $O(mn)$ where m and n are the lengths of the input sequences. Probalign's memory requirements can be improved to $O(mn^{1/2})$ with a 1.5 factor slowdown using memory reduction techniques used for HMM-based alignment programs [55]. This is part of planned future work.

Finally, it remains to be seen if Probalign partition function posterior probabilities demonstrate the same level of improvement seen here for the profile-sequence alignment and profile-profile alignment problems. The utility of profiles, however, is limited when unknown and unalignable genomic flanks are present or the family alignment is not rich or accurate enough. In that case, the current Probalign implementation offers a viable solution as demonstrated.

3.4 Conclusion

This report represents the first examination of the Probalign alignment algorithm to search for RNA homologs within much larger genomic segments using partition function posterior probabilities. It shows that the method does much better than the widely used SSEARCH and BLAST programs. Furthermore, the Probalign mean posterior probability (which has previously been discussed as a possible metric to assess alignment quality, but never studied carefully) has been shown to be a better indicator of alignment quality than the standard SSEARCH Z-score.

3.5 Methods

Benchmark

This work began by extracting all 26 RFAM [49] seed alignments with known published RNA secondary structures and average pairwise sequence identity of at most 60%. During the benchmark construction process four families fail to meet length and uniqueness criteria (see below); this subsequently leaves us with 22 families in the end. At the time of the writing of this report, RFAM version 7.0 was the most recent release. Sequence identity is measured only in regions of known secondary structure, which are generally more reliably aligned than the rest. The 60% threshold has previously been identified as a cutoff for hard RNA alignment cases [39] and so this endeavor focuses specifically on this region. The following three main steps are used to construct the benchmark from the initial 26 families:

1. ***Pairwise RNA-RNA alignments***: For each of the initial 26 RFAM seed multiple alignments, a maximum of 350 pairwise alignments are randomly selected. In families where there are less than 350 total pairs, all are considered.

2. ***Construction of genomic flanks***: Every RNA sequence in RFAM seed is cross-linked to a genomic sequence in EMBL [32]. For each pairwise alignment produced above, one of the RNA sequences is randomly selected and real genomic flanks from EMBL (version r88) are attached to each end of the RNA. Each genomic flank is truncated to 7500 base pairs on either end. Since the largest RNA sequence is at most 1000 nucleotides long, the maximum size of each genomic sequence is 16,000. This gives RNA genome alignments where the RNA sequence can be considered as a query and the aligned homologous RNA is the target "hidden" in the genome. In order to make the dataset challenging enough, datasets where the genomic sequence is shorter than 5000 nucleotides are excluded.

3. ***Alignment uniqueness***: The attached genomic flanks may contain additional related RNAs of the same family as the query and the target (to which the flanks were attached). This means that two different correct alignments are possible. To keep

things simple, such datasets are excluded and it is ensured that each query-target alignment is unique. For each dataset, a profile was built from the RFAM family alignment annotated with consensus secondary structure using the cmbuild program of the INFERNAL suite. The cmsearch program of the INFERNAL suite was then run on the genomic sequence of the dataset and it was excluded entirely from the benchmark if more than one hit above a bit score of 30 was reported.

4. The pruning process yielded a total of 3,429 pairwise alignments distributed (unequally) among 22 RNA families. As mentioned earlier, all the datasets in four families failed to meet the length and uniqueness criteria just described. Subsequently, in the end, 22 families are left. The 22 families and their characteristics can be found in Appendix B.

Adding Genomic Flanks to Query RNA

To simulate practical conditions where the exact 5' and 3' ends of ncRNAs are unknown, each dataset in the benchmark was taken and three similar versions were produced. However, in each of the three versions, real 5' and 3' genomic flanks of size 50, 100, and 150 nucleotides were added to the query RNA of each dataset. By cross-referencing each RNA sequence to the original genomic version in EMBL it was possible to obtain proper real genomic flanks and hence simulating artificial ones was not needed. Subsequently, the size of the benchmark increased four-fold from 3,429 to 13,716. Gaps were removed from each alignment. The flanked query and target genomic sequences were used as input to each program.

The full benchmark is available online [56]. Also available at the website are the RFAM family alignments from which the benchmark was created, training datasets (see below), and false positive datasets used for discrimination tests (described below).

3.6 Alignment Programs and Parameters

Training Data

This report used a subset of the benchmark with query RNA flanks of size 100 nucleotides for training the program parameters. For each of the 22 divergent families, 25 random datasets were selected. If the family contained a total of less than 25 pairwise alignments, all were included in the training set. The final training set contained 498 pairwise alignments and can be found on the website for this report [56].

Probalign

A modified version of the Probalign beta 1.0 program more attuned to local alignment was used. Two modifications were made to the partition function matrices. They follow from analogous standard dynamic programming recursions for local alignment and can also be found in Muckstein et. al. [47]. First, 1 is added in the calculation of the match partition function matrix: $Z^M_{i,j} = \left(1 + Z^M_{i-1,j-1} + Z^E_{i-1,j-1} + Z^F_{i-1,j-1}\right) e^{s(x_i,y_j)/T}$. Second, the total partition function value is set to $Z = 1 + \sum_{i,j} Z^M_{i,j}$. The initial values of the Z-matrices also need to be set appropriately in line with the two changes. However, since zero end-gaps are used, this is automatically set. A more detailed description of the partition function matrices and notation is given in Appendix D. Probalign returns one alignment of the complete query against the genomic sequence. However, a Perl script [56] is provided to produce multiple alignments of significant mean posterior probability. This script produces multiple alignments of the query against the genomic sequence by removing the aligned portion of the genome to the query and realigning the remainder to the query until the mean posterior probability is zero. In other words, all hits above zero probability are reported. This parameter can easily be modified in the script. Only the top hit in the experiments are evaluated. The SSEARCH +5/-4 scoring matrix is used for Probalign. The gap open and gap extension penalties as well as the thermodynamic temperature are optimized on the training benchmark. The modified Probalign program is available as standalone code [56].

BLAST

The bl2seq program (current version 2.2.16) of the NCBI BLAST suite is used in the experiments. In the terminology of this dissertation work, BLAST represents the bl2seq program of the suite. BLAST returns local alignments that may not include the entire query. In order to measure the error correctly, the entire query is required to be aligned to the genomic sequence (see Prediction Error Sub-section below). This is accomplished by extending the local alignment in either direction until the full query is aligned to the genomic sequence. Only the highest E-value BLAST hit is evaluated. The performance of the second hit outputted on each dataset was tested and it was found to have a much worse error than the first. This is expected since each pairwise alignment in the benchmark is unique. BLAST gap parameters were optimized using both its default scoring matrix (+3/-1) and +5/-4 (the same one as used in SSEARCH). In order to avoid excessive scenarios where BLAST does not return an alignment, the minimum word size is set to 4. The +5/-4 matrix is used for BLAST since it performs better than the default (both with optimized parameters) on the training benchmark.

SSEARCH

SSEARCH release version 3.4t26 is used in the experiments. SSEARCH is a local alignment program and may not contain the entire query aligned to the genome (necessary for correct error computation). This problem can be fixed using the same BLAST treatment described above. With the -a option, however, SSEARCH returns alignments of both query and genome sequence in their entirety. In this case the accuracies are found to match those calculated otherwise, which is by fixing the alignments if necessary. Thus, without loss of any accuracy SSEARCH is run with -a enabled. The SSEARCH gap open and gap extension penalty parameters are optimized on the training benchmark. Like BLAST, the second SSEARCH hit was found to be significantly much worse off than the first one.

ClustalW

ClustalW version 1.83 is used for the experiments. ClustalW, like Probalign, returns one global alignment of the complete query against the genomic sequence. The terminal gap (end-gap) penalties are set to zero. ClustalW gap parameters are optimized on the default ClustalW scoring matrix of +10/-9 and the SSEARCH +5/-4 scoring matrix. However, the ClustalW default matrix optimal gap parameters perform better than the optimized +5/-4 matrix.

HMMER

HMMER version 2.3.2 is used in the experiments. This was the current version at the time of writing this report. HMMER is designed for profile-based search that requires family alignments. Since the goal of this report is to study query RNA genomic search, particularly for divergent and hard cases where family alignments are not reliable, the single sequence RNA query is used for constructing the HMMER model. The hmmbuild program of the HMMER suite is used to build local alignment models (with the -f and -s options) and global alignment models (with the -g option). The hmmsearch program is then used on the training benchmark and on each of the three models to search the genomic sequence for homologs of the query RNA. The local alignment -f and -s models are found to be equally best performing. -f is used in the experiments. Like BLAST and

SSEARCH, HMMER local alignments may not contain the full query aligned to the genome. Therefore, it is fixed in the same manner described above in the BLAST option.

Table 3.5 provides all parameters used in the four non-model based methods. The HMMER parameters are estimated from the single sequence profile specific to each dataset and therefore are not included in Table 3.5. The exact command line options used for running the programs are listed in Appendix C.

Table 3.5 Description of Optimized Parameters Derived for Each Method used Herein

Method	Scoring matrix	Gap opening penalty	Gap extension penalty
Probalign	+5/-4 (T = 7)	32	2
SSEARCH	+5/-4	10	4
ClustalW	+10/-9	13	6
BLAST	+5/-4	8	6

3.7 Alignment Quality Measures

Probalign Mean Posterior Probability

The Probalign mean posterior probability is defined by Equation 3.1. $P(x_i \sim y_j)$ is the posterior probability of the i^{th} nucleotide of sequence x aligning to the j^{th} nucleotide of sequence y. More details about how this is computed and the Probalign method in general can be found in Appendices B, C, and D.

$$\text{Probalign mean posterior probability} = \frac{\sum_{x_i \neq -, y_j \neq -} P(x_i \sim y_j)}{(\text{\#aligned nucleotides with non-zero posterior probability})} \quad (3.1)$$

SSEARCH Normalized Z-score

The SSEARCH Z-score and E-value are standard statistical measures of alignment reliability [57, 58]. The Z-score can be compared across different sequence pairs [59]. The normalized Z-score is used as a predictor of alignment quality. The normalized Z-score is the standard Z-score divided by the number of aligned nucleotides in the local alignment. This is found to produce a much better ROC analysis than the raw Z-score and the normalized and raw E-value.

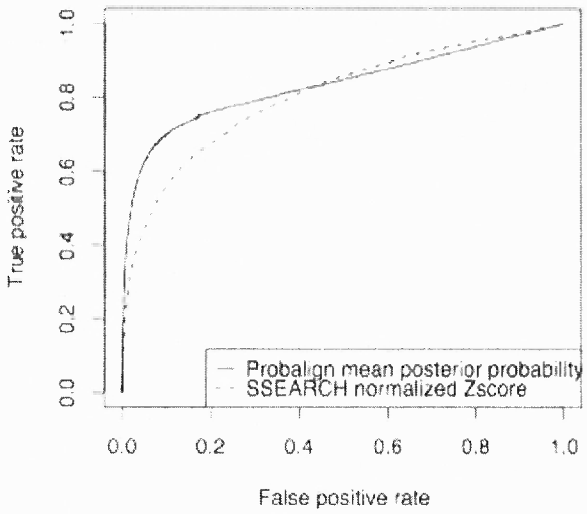


Figure 3.1 ROC curves for Probalign mean posterior probability and SSEARCH normalized Z-score. To construct this curve, a set of false hits were added to the dataset by replacing each genomic sequence in each dataset of the benchmark with a randomly selected one from a benchmark dataset of a different RNA family. The ROC analysis clearly demonstrates that Probalign is better able to discriminate true from false alignments.

False Positive Datasets

A set of false positives were created in order to measure the prediction accuracy of the above two measures. For each dataset in the benchmark, a false positive one is created by replacing the genomic sequence with one selected from a different random dataset. Now, each false positive dataset contains a query RNA and a genomic sequence containing a target RNA from a different family. It is expected that any alignment reliability measure will have a low value on these datasets. These datasets are available online [56].

3.8 Measure of Accuracy and Statistical Significance

Prediction Error

The goal of this work is to find out how much of the target RNA (which lies in the genomic sequence) is aligned to the query, excluding the query flanks. As described above, for BLAST, SSEARCH, and HMMER, all of which return local alignments, the query-genome alignment is extended in both directions until the entire query, but not its flanks, is matched to the genomic sequence. This improves sequence coverage, reduces the false negative rate, and also allows a fair comparison to Probalign and ClustalW, both of which return global alignments of the entire sequences. For each method, the part of the genomic sequence aligned to the query in its alignment is taken; the false positives are measured as the number of nucleotides in this region that are not in the target RNA. Similarly, the false negatives are measured as the number of nucleotides in the target RNA that are not in the genomic region aligned to the query (in the method estimated alignment). See Figure 3.2 for a visual description of the false positives and false negatives. The false positive and false negatives are normalized by the size of the genomic region aligned to the query in the computed alignment and the size of the target RNA respectively. The normalized false positive and false negatives can now be expressed as a percentage between 0 and 100. The error, also expressed as a percentage, is measured as the average of the normalized false positive and false negative.

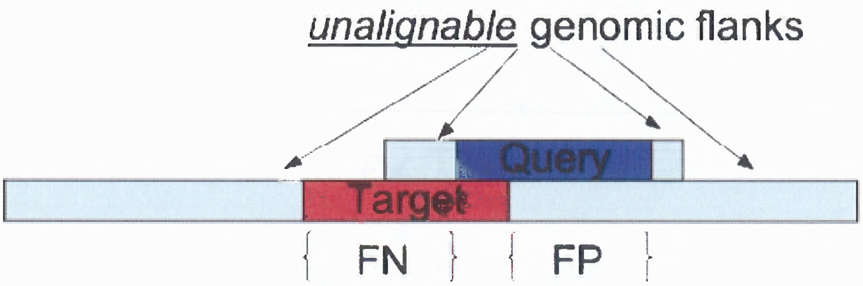


Figure 3.2 A cartoon of false positive and false negative situations for a query-target alignment.

Statistical Significance

Statistically significant performance differences between the various alignment methods are calculated using the Friedman rank test [60]. This is a standard measure used for discriminating alignments in benchmarking studies [5, 34]. Roughly speaking, lower P-values coincide with reduced likelihoods that the ranking differences are due to chance. This report considers P-values below 0.05 (a standard cutoff in statistics) to be statistically significant.

Correlation With True Hits and True Accuracy

A ROC analysis [61] is conducted to study how well the Probalign mean posterior probability and the SSEARCH Z-score can predict the quality of the alignment. An ROC curve plots the true positive rate (y-axis) against the false positive rate (x-axis). The area under the curve is an indicator of overall accuracy of the classifier. All ROC area under curve values are normalized to 1 with higher areas indicating higher accuracy. The Probalign mean posterior probability and the SSEARCH normalized Z-scores are treated as classifiers for a true or false hit.

CHAPTER 4

WEBSERVERS: EPROBALIGN AND PROBALIGN-LOCAL

As discussed earlier, Probalign computes maximal expected accuracy multiple sequence alignments from partition function posterior probabilities. To date, Probalign is among the very best scoring methods on the BAliBASE, HOMSTRAD and OXBENCH benchmarks. eProbalign and Probalign-local, which are web/online implementations of Probalign and Probalign-RNAgenome respectively, are described in the following sections. The eProbalign web server doubles as an online platform for post-alignment analysis. The core of the post-alignment functionality is the Probalign Alignment Viewer applet, which provides users a convenient means to manipulate the alignments by posterior probabilities. The Alignment Viewer can also be used to produce graphical and text versions of the output. The eProbalign web server and underlying Probalign source code is freely accessible at http://probalign.njit.edu. The Probalign-local web server is available online at http://probalign.njit.edu/local.

4.1 Introduction

Multiple sequence alignments are frequently employed for analyzing biomolecular sequences. Their application spans a wide range of problems such as phylogeny reconstruction, protein functional site detection, and protein and RNA structure prediction [29]. The research literature is abundant with programs and benchmarks for multiple sequence alignment, particularly for protein data. Traditionally, ClustalW [30] is the most popular program used for multiple sequence alignment; while BAliBASE [31] is likely the most commonly used benchmark of protein alignments. MAFFT, Probcons and Probalign are recent alignment strategies that are among recent programs with the highest accuracies on BAliBASE and other common benchmarks (i.e., HOMSTRAD [32] and OXBENCH [33]). Both Probcons [9] and Probalign [10] compute maximal expected accuracy alignments using posterior probabilities.

In Probcons, posterior probabilities are derived using an HMM whose parameters that have been estimated via supervised learning on BAliBASE unaligned sequences. Probalign, which is largely based on the Probcons scheme, derives the posterior probabilities from the input data by implicitly examining suboptimal (sum-of-pair) alignments using the partition function methodology for alignments (see [10] for a full description of the algorithm). Probalign alignments have been shown to have a statistically significant improvement over Probcons, MAFFT [8] and MUSCLE [34] on all three alignment benchmarks introduced above [10].

eProbalign is a web server that automatically computes Probalign alignments. It also provides a convenient platform to visualize the alignment, generate images, and manipulate the output by average column posterior probabilities. The average column posterior probability (which is discussed further below) can be considered a measure of column reliability where columns with higher scores are more likely to be correct and perhaps biologically informative.

Probalign-local web-server is an online implementation of Probalign-RNAgenome and has the same specifications as the Probalign-RNAgenome program described in Chapter 3.

4.2 Input Parameters

eProbalign takes as input unaligned protein or nucleic acid sequences in FASTA format. eProbalign checks the dataset to make sure that it conforms with IUPAC nucleotide and amino acid one letter abbreviations. White space between residues/nucleotides in the sequences are stripped and the cleaned sequences are passed on to the queuing system. The user can specify gap open, gap extension, and thermodynamic temperature parameters on the eProbalign input page (Figure 4.1). The input page provides a brief description of the parameters (help link) and links to the standalone Probalign code with publication and datasets.

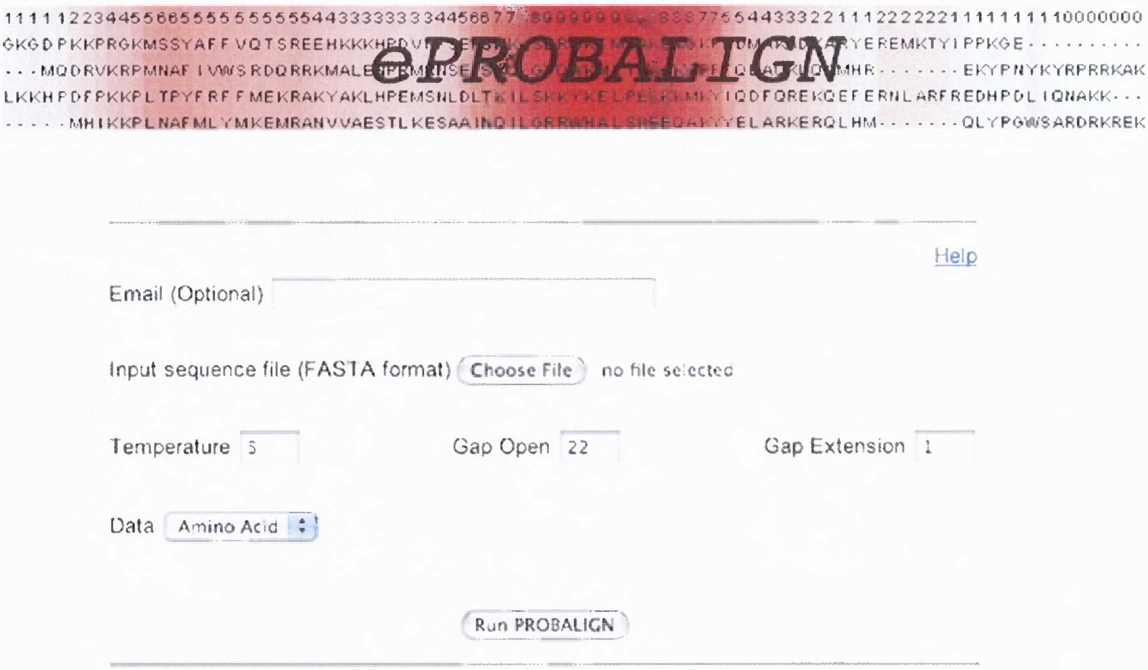
The three Probalign parameters on the input page are used for computing the partition function dynamic programming matrices from which the posterior probabilities are derived. This is the same as computing a set of (suboptimal) pairwise alignments (for every pair of sequences in the input) and then estimating pairwise posterior probabilities by simple counting. The thermo dynamic temperature controls the extent to which suboptimal alignments are considered. For example, all possible suboptimal alignments would be considered at infinite temperature, whereas only the single best would be used at a temperature of zero. The affine gap parameters are used for the pairwise alignments. Subsequently, Probalign computes the maximal expected accuracy alignment from the posterior probabilities in the same way that Probcons does [9].

4.3 Output and Alignment Column Reliability

The eProbalign output provides three options for viewing and analyzing the alignment (Figure 4.2). The alignment can be viewed in (i) FASTA text format, (ii) pdf graphical format, and (iii) the Probalign Alignment Viewer (PAV) applet (Figure 4.4). Each column of the alignment in the pdf file and in the applet is colored in a shade of red according to the average column posterior probability. Bright red indicates probability close to one whereas white indicates close to zero (see Figure 4.3 for an example on a real BAliBASE dataset).

The average column posterior probability is defined as the sum of posterior probabilities of all pairwise residues in the column normalized by the number of comparisons [9]. The top row of the alignment in the pdf and applet displays the average column posterior probabilities multiplied by ten and floored to the lower integer (Figure 4.3). For example, a score of 1 indicates that the probability is between 0.1 and 0.2.

The Probalign Alignment Viewer is a Java applet that provides basic manipulation of the alignment. Basic Java and browser requirements to use the applet are listed on the output page. With the applet the user can opt to view and save the alignment with column posterior probabilities above any specified threshold. This has the benefit of "cleaning up" the alignment by column posterior probabilities, which is unique to eProbalign. The applet also displays posterior probabilities of all columns in a separate window if desired (Figure 4.4) and provides options to switch between the gapped and ungapped versions of the alignment.



Email (Optional)

Input sequence file (FASTA format)

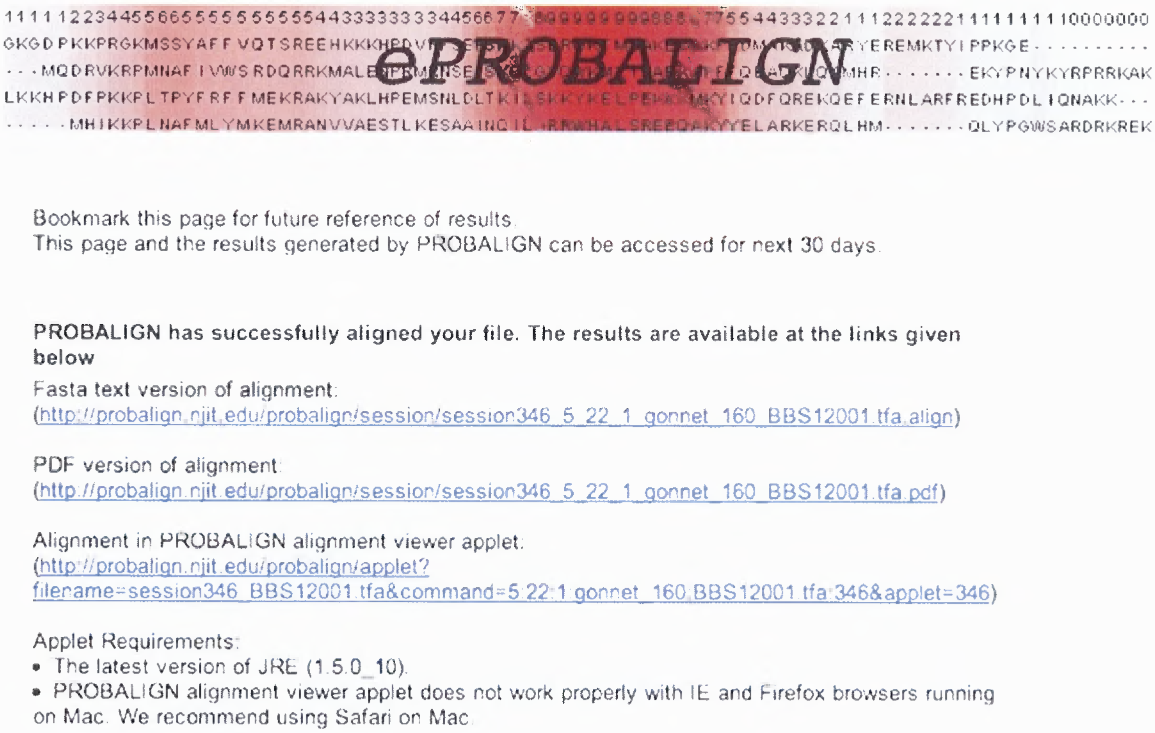
Temperature 5

Gap Open 22

Gap Extension 1

Data Amino Acid

Figure 4.1 eProbalign input page.



Bookmark this page for future reference of results.
This page and the results generated by PROBALIGN can be accessed for next 30 days.

PROBALIGN has successfully aligned your file. The results are available at the links given below

Fasta text version of alignment:
(http://probalign.njit.edu/probalign/session/session346_5_22_1_gonnet_160_BBS12001.tfa.align)

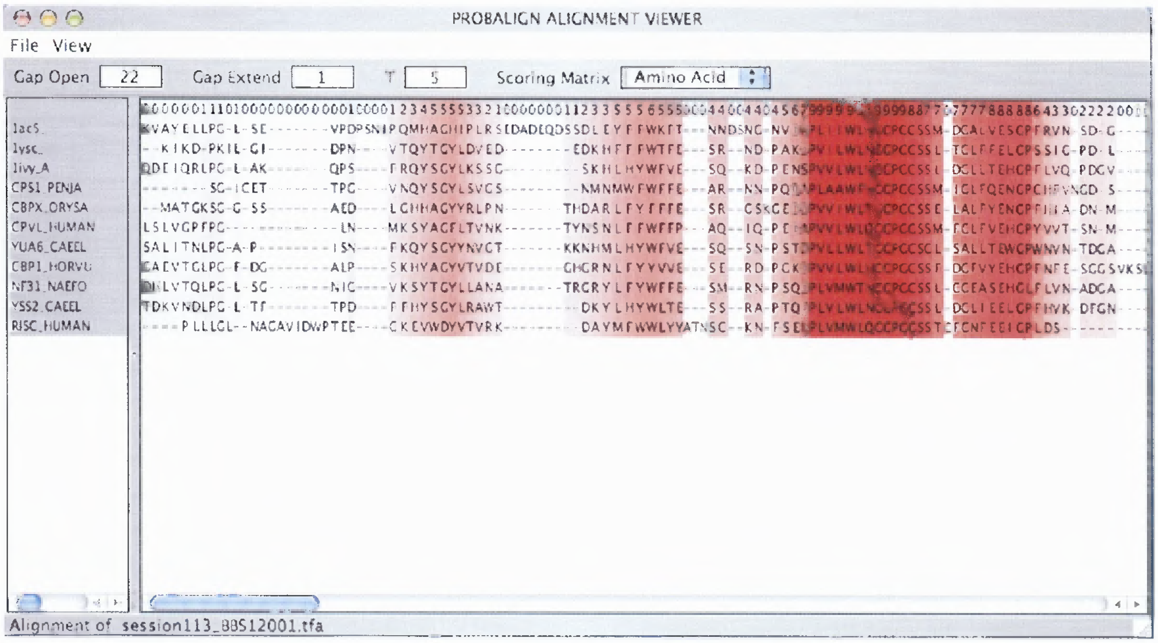
PDF version of alignment:
(http://probalign.njit.edu/probalign/session/session346_5_22_1_gonnet_160_BBS12001.tfa.pdf)

Alignment in PROBALIGN alignment viewer applet:
(http://probalign.njit.edu/probalign/applet?filename=session346_BBS12001.tfa&command=5:22:1:gonnet_160.BBS12001.tfa:346&applet=346)

Applet Requirements:

- The latest version of JRE (1.5.0_10).
- PROBALIGN alignment viewer applet does not work properly with IE and Firefox browsers running on Mac. We recommend using Safari on Mac.

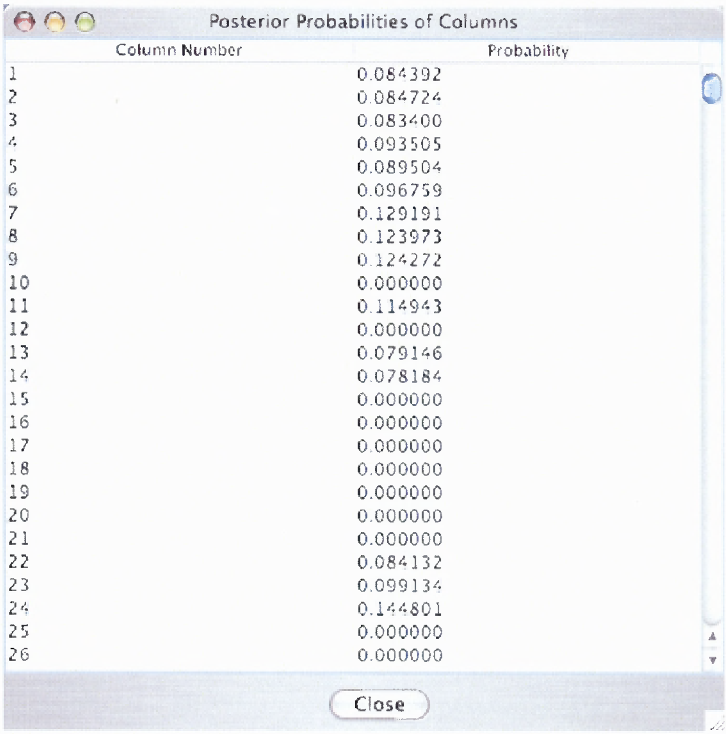
Figure 4.2 eProbalign output page indicating results are done.



Gap Open 22 Gap Extend 1 T 5 Scoring Matrix Amino Acid

Alignment of session113_BBS12001.tfa

Figure 4.3 Probalign Alignment Viewer applet.



Column Number	Probability
1	0.084392
2	0.084724
3	0.083400
4	0.093505
5	0.089504
6	0.096759
7	0.129191
8	0.123973
9	0.124272
10	0.000000
11	0.114943
12	0.000000
13	0.079146
14	0.078184
15	0.000000
16	0.000000
17	0.000000
18	0.000000
19	0.000000
20	0.000000
21	0.000000
22	0.084132
23	0.099134
24	0.144801
25	0.000000
26	0.000000

Figure 4.4 Posterior probability of each column.

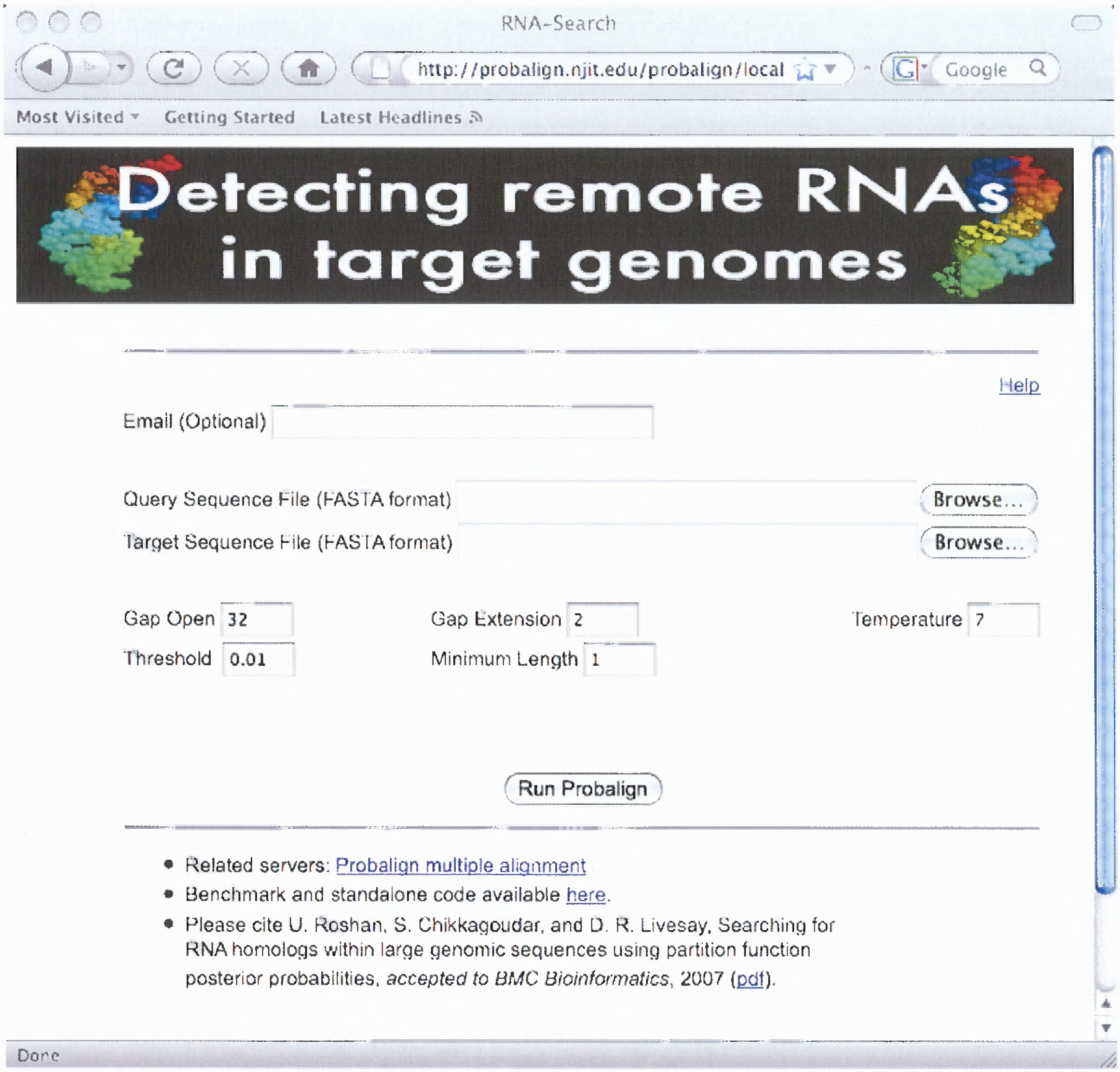
4.4 Server Implementation

eProbalign implements a first-in/first-out queuing system that receives requests for Probalign alignments and processes them accordingly. At most, eProbalign will run two Probalign jobs at once, and it will periodically check the queue for new requests. Alignments that take longer than some defined time limit (10 hours at the time of writing of this report) are stopped and the user is advised to download and run the standalone version. This time limit will be increased as the server hardware is upgraded.

Probalign-local web server was implemented in CGI/Perl. Figures 4.5, 4.6 and 4.7 are snapshots of it.

4.5 Scalability

Currently, at NJIT, eProbalign is installed on a dual processor 2.8GHz Intel Xeon machine with 2GB RAM. With these settings, eProbalign can usually align datasets of up to 20 sequences within one minute. Most BAliBASE 3.0 datasets from RV11 and RV12 also finish within one minute. eProbalign has also been tested on large datasets (in number and length of sequences) from BAliBASE RV30 and RV40 classes. BB30029 and BB30008 from RV30 contain 98 and 36 sequences with lengths from 431 to 852 and 400 to 1155, respectively, and BB40002 from RV40 contains 55 sequences with lengths ranging from 58 to 1502. When the server is idle, eProbalign finished in about 20 minutes on BB30008, 55 minutes on BB30029, and 30 minutes on BB40002. Results may take longer to finish when the server queue is full and multiple jobs are running simultaneously. However, the effect of parallel jobs will diminish as the server moves to a bigger machine in the near future.



Help

Email (Optional)

Query Sequence File (FASTA format)

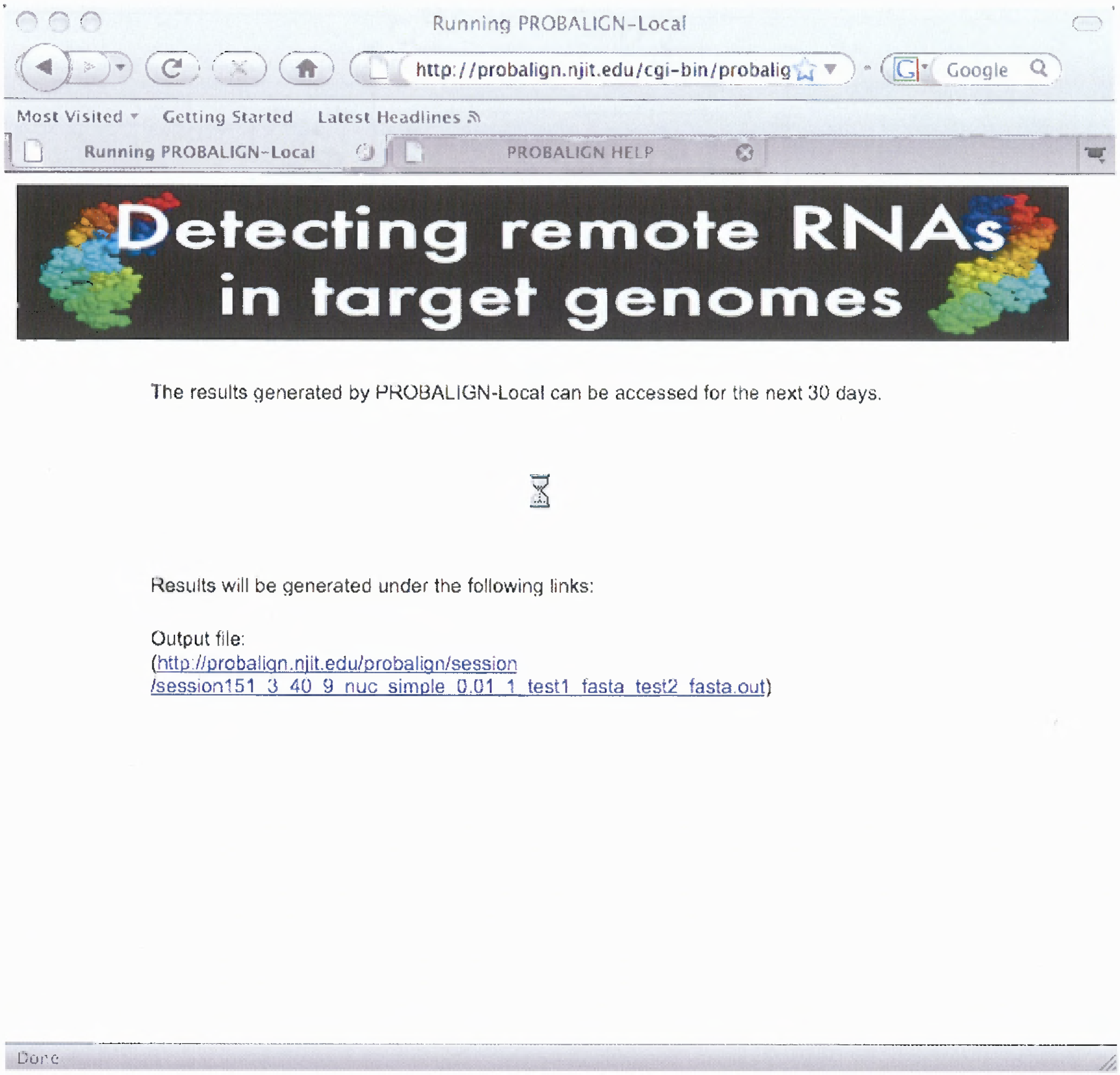
Target Sequence File (FASTA format)

Gap Open 32 Gap Extension 2 Temperature 7

Threshold 0.01 Minimum Length 1

- Related servers: Probalign multiple alignment
- Benchmark and standalone code available here.
- Please cite U. Roshan, S. Chikkagoudar, and D. R. Livesay, Searching for RNA homologs within large genomic sequences using partition function posterior probabilities, *accepted to BMC Bioinformatics*, 2007 (pdf).

Figure 4.5 Probalign-local input page.

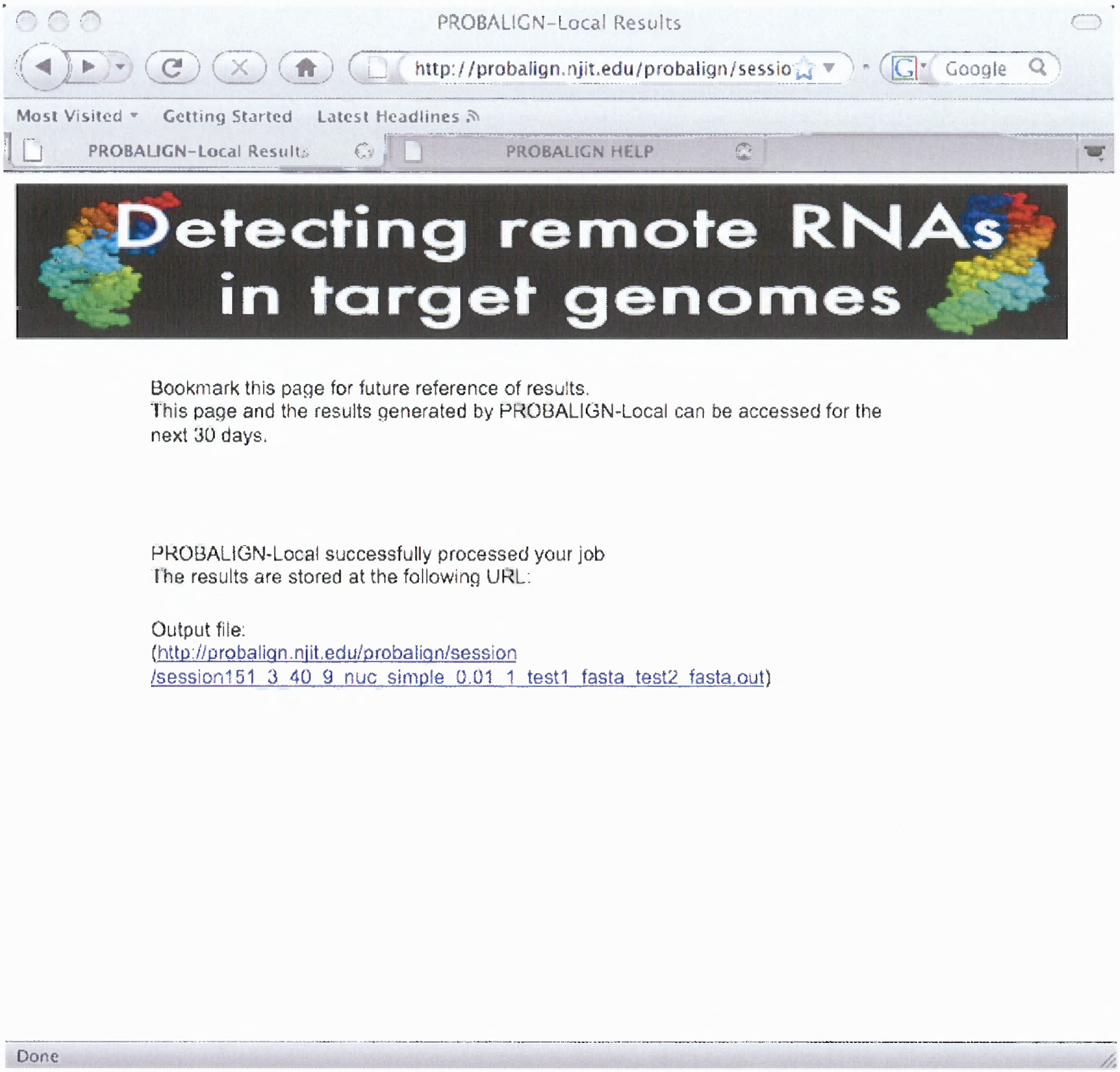


The results generated by PROBALIGN-Local can be accessed for the next 30 days.

Results will be generated under the following links:

Output file:
(http://probalign.njit.edu/probalign/session
/session151_3_40_9_nuc_simple_0.01_1_test1_fasta_test2_fasta.out)

Figure 4.6 Probalign-local intermediate page indicating that result is being computed.



Bookmark this page for future reference of results.
This page and the results generated by PROBALIGN-Local can be accessed for the next 30 days.

PROBALIGN-Local successfully processed your job
The results are stored at the following URL:

Output file:
(http://probalign.njit.edu/probalign/session
/session151_3_40_9_nuc_simple_0.01_1_test1_fasta_test2_fasta.out)

Figure 4.7 Probalign-local output page indicating that results are ready.

CHAPTER 5

A HYBRID SUPPORT VECTOR MACHINE STRATEGY FOR RANKING SNPS IN GENOME-WIDE ASSOCIATION STUDIES

In genome wide association studies, the goal is to rank SNPs such that true associated ones are placed at higher positions than false ones. The support vector machine (SVM) provides a discriminative alternative to the widely used chi-square statistic. This chapter describes a hybrid strategy that combines the chi-square statistic with the support vector machine and studies its performance on simulated data and the Wellcome Trust Case Control Consortium (WTCCC) studies. The following sections will show that this hybrid strategy ranks causal SNPs in simulated data significantly higher than the chi-square test and SVM alone. It will also be shown that this novel strategy ranks previously replicated SNPs and associated regions (where applicable) of type 1 diabetes, rheumatoid arthritis, and Crohn's disease higher than the chi-square, SVM, SVM-RFE, and the HMM SNP rankings. In WTCCC studies with low signal strength such as type 2 diabetes there is no advantage to this hybrid method. Finally, it will be shown that this hybrid strategy yields an economical set of SNPs that predict disease risk more accurately than previously replicated SNPs and top ranked SNPs in the chi-square and SVM ranking for type 1 diabetes and arthritis as measured by the area under curve of the widely used composite odds ratio score.

5.1 Introduction

Genome-wide association studies aim to identify genetic variants associated with disease, drug response, and various phenotypes [97]. The standard method of ranking SNPs from genome-wide association studies is the two or one degree of freedom chi-square test [74, 97]. This is referred to as the chi-square test from hereon (with two degrees of freedom).

Previous studies have examined the performance of the chi-square statistic in ranking SNPs [81, 99], proposed techniques to improve the rankings under two-stage designs [80] and to correct for overestimated significance values and apply the false discovery rate control method thereafter [79, 102]. Other approaches instead of chi-square have also been proposed for ranking SNPs. These include the trend test [81, 100], Bayes factors [97], random forests [83, 85], a penalized maximum likelihood (ML) approach [71], and a hidden Markov model (HMM) based method [96]. None of these except for the HMM method have reported significant improvements over the chi-square ranking of SNPs. In fact, Bayes factors rankings were been found very similar to chi-square as reported by Wellcome Trust Case Control Consortium [97].

The support vector machine (SVM) provides a discriminative alternative to the chi-square statistic for feature selection. Although originally designed for classification it can also be used to rank features and has been studied extensively for the gene selection problem [69]. The intuition behind ranking SNPs by an SVM lies in the SVM discriminant vector w itself. The SVM classifies a given data point x by taking the dot product of w with x plus a bias term. Since the larger entries of w have a greater influence on the dot product than the smaller ones it is intuitive to rank the features by their entries in w.

This does not mean that the entries of w are guaranteed to assign higher weights to causal variables, as shown recently in simulation and theoretically [70, 93]. As a further validation of the results in Statnikov et al. [93], this work shows that the SVM if applied to a genome-wide association study does not necessarily rank known replicate SNPs or causal ones in simulated data at higher positions than the baseline chi-square.

Aside from the basic SVM discriminant there is the highly popular SVM recursive feature elimination (RFE) algorithm [69] for feature selection. It computes the SVM classifier, ranks features by their entries in the SVM discriminant, eliminates features with the lowest entries, and reiterates this process until a desired number of features remain. It has been studied extensively for ranking genes given their expression data. For the problem of ranking SNPs, however, the subsequent sections will show that it does not perform better than chi-square on real data.

In this study, a hybrid strategy that combines the support vector machine with the chi-square statistic is proposed. The top r ranked SNPs in the chi-square ranking are selected and re-ordered with the SVM discriminant with a specified value of C, the SVM loss-complexity tradeoff parameter. The hybrid strategy automatically determines r and C.

Before comparing the performance of the hybrid strategy on real data to other methods, it is compared to the baseline chi-square and SVM separately on simulated data where the causal allele is known. Genome-wide association studies are simulated with same LD structure as the HapMap CEU genotypes, 1,000 case and 1,000 controls, and .01 disease prevalence. Relative risks of 1.25, 1.5 and 2 with ten and 15 causal alleles and several thousand non-causal SNPs are examined. For each setting of relative risk and number of causal alleles, 50 studies are generated. This work finds that causal alleles are placed significantly higher in the hybrid SNP rankings compared to the one given by chi-square and SVM. Furthermore, the improvements are statistically significant on datasets with 15 causal alleles and relative risk of 2 with ten causal alleles.

This work then compares the rankings of previously replicated SNPs of type 1 diabetes, arthritis, Crohn's disease, and type 2 diabetes in the SNP orderings of Wellcome Trust Case Control Consortium (WTCCC) studies given by the hybrid strategy, chi-square, SVM, SVM-RFE, and the HMM. A recent publication [66] containing replicated SNPs for the four diseases as well as a curated table of associated regions at the Type 1 Diabetes Consortium [63, 72] was referred to for obtaining type 1 diabetes associated regions. The hybrid strategy ranks most replicated SNPs higher than all other methods on all diseases except for type 2 diabetes, which has the weakest signal of all four. It also ranks SNPs from known type 1 diabetes associated regions higher than the other methods without the expense of additional false positives SNPs (i.e., those that do not belong to any known associated region), thus making such regions detectable by examining only a few top ranked SNPs.

Finally, this report compares the accuracy of the industry standard disease risk estimator as a function of top ranked hybrid SNPs, previously replicated SNPs, and top ranked chi-square and SVM SNPs on WTCCC studies of the four diseases. It finds an improvement of 2% with top ranked hybrid SNPs in type 1 diabetes and arthritis, 1% in Crohn's disease, and none in type 2 diabetes (the last two diseases have relatively lower signal strength). Furthermore, the improvement in type 1 diabetes and arthritis is given by an economical set of at most 37 top ranked hybrid SNPs.

This report concludes that the hybrid strategy ranks causal and replicated SNPs better than chi-square and SVM separately and that an economical set of top ranked ones predict disease risk more accurately than top ranked chi-square and SVM ones as well as previously replicated ones on studies with moderate to high signal strength. One can expect this strategy to be more useful as larger studies with deeper sequencing and relatively stronger signal strengths become available. Perl scripts and C programs are provided at http://www.cs.njit.edu/usman/SVMSNP for reproducing the results and running the SVM strategy on a given study. In the remainder of the report, the hybrid approach is described, and detailed experimental results are provided.

5.2 Methods

Background on the 2 degree-of-freedom chi-square test, support vector machine, and the composite odds ratio score is provided in Appendix E. The hybrid SNP ranking strategy and details on the real and simulated data are presented in the subsequent sub-sections.

5.2.1 Ranking SNPs with a Hybrid SVM Strategy

The strategy is a simple one: select the top r SNPs in the chi-square ranking and re-order them with the SVM discriminant with a specified value of C, the SVM loss-complexity tradeoff parameter. However, the selection of r and C are critical to the performance of this strategy. r and C are selected such that SNPs that best classify case and control are placed at high positions.

- **Input**: n case and control samples each with m SNP genotypes, SNP identifiers $\{s_1, s_2, ..., s_m\}$, and the set of values of r and C from which to select the optimal one
- **Output**: Optimal values of r and C, the SVM discriminant vector $w = (w_1, ..., w_r)$, vector p such that $|w_{p_1}| \geq |w_{p_2}| \geq ... \geq |w_{p_r}|$, and corresponding ranking of input SNPs $s_{p_1}, s_{p_2}, ..., s_{p_r}$.
- **Method**:
  a. Convert the input genotypes into an encoded matrix of 0, 1, and 2's by a standard encoding [89].
  b. Produce ten random training-validation subsets where 90% of case and controls are in training and remaining 10% in validation.
  c. For each value of r to consider:
     i. Compute the chi-square ranking of SNPs using the training set and obtain the top r ranked ones.

ii. For each value of C to consider:

1. Compute the SVM discriminant w with the loss-complexity parameter set to C and reorder the SNPs. A method to obtain a SNP ordering using w is described below.

2. Select the top t SNPs in the SVM ordering, where $t = 5,10,15,25,30,35,40,45,50$ and compute classification error on the validation set with the non-parametric nearest centroid classifier [62]. In other words, the goal is to find suitable values of C and r that move discriminative SNPs to high ranks.

3. Store the error with the given value of C, r, and t.

d. Return the value of C, r, and t with minimum average error across the ten training-validation sets and compute the SVM ranking with these values of C and r.

The SVM discriminant w is computed with the *SVM-light* software package [75]. An ordering of the SNPs can be obtained using the absolute value of the entries of w. The i^{ith} entry of w represents the weight of the i^{ith} SNP in the SVM classifier. Let $w = (w_1, ..., w_r)$ and $|w| = (|w_1|, ..., |w_r|)$. Now consider the entries of $|w|$ in sorted descending order. This ordering is denoted by the vector p such that $|w_{p_1}| \geq |w_{p_2}| \geq ... \geq |w_{p_r}|$. An ordering is obtained on the input SNP identifiers $s_{p_1}, s_{p_2}, ..., s_{p_r}$ using p. This gives us the SNP ranking.

The SVM baseline discriminant can be replaced with a different one such as a regularized risk minimizer [94] (motivated by SVMs [92]) or a discriminative dimensionality reduction method, such as the weighted maximum margin discriminant [101]. The SVM discriminant is selected because it is supported by powerful theoretical and empirical performance.

In summary, this hybrid strategy searches for the ranking such that SNPs at high positions minimize the nearest centroid classification error on the validation set. There is no guarantee they will contain causal or replicated ones. However, in the simulated and real data experimental results presented below it is clear that this turns out to be the case at least in studies with moderate to high signal strength.

The implementation of this strategy is a combination of Perl scripts and C programs. It is available for download at http://www.cs.njit.edu/usman/SVMSNP/. In practice, the running time of the hybrid strategy is fast, thanks to efficient implementations of baseline programs. The two degree of freedom chi-square test is implemented with 2×3 contingency tables in C and the nearest centroid classifier in Perl. The SVM-light software package is used and is very fast in practice. To give an idea of the running time on an AMD Opteron 64bit machine, the hybrid strategy takes 20 minutes to finish on a simulated study with approximately 30,000 SNPs and 2,000 subjects for $r = 100$ and C optimally selected from the set .1 through 10^{-7} in increments of 10^{-1} (total of seven values of C). Note that this is the total running time and it includes computing the chi-square ranking separately for ten training validation splits of the input dataset. However, this implementation could be made more efficient by combining it into one C program.

5.2.2 Datasets

WTCCC Studies [97]

The Wellcome Trust Case Control Consortium (WTCCC) provides two sets of controls and one set of cases each for type 1 diabetes, rheumatoid arthritis, Crohn's disease, and type 2 diabetes [97]. It also provides case subjects for bipolar disorder, hypertension, and coronary artery disease. However, they are omitted from this study because their signal strength is similar to type 2 diabetes and fewer replicated SNPs are catalogued for them in comparison to the other four.

All SNPs and samples are removed from cases and controls that are specified as problematic by the WTCCC. This leaves 1,480 individuals from the 1,958 British Birth Cohort, 1,458 from the UK Blood Service Control Group, 1,963 cases for type 1 diabetes, 1,860 for arthritis, 1,748 for Crohn's disease, and 1,924 for type 2 diabetes. The two control sets are combined with each case set and all SNPs with greater than 1% missing entries are removed. Using the plink software package [90, 91], all SNPs that deviate from the Hardy-Weinberg equilibrium with p-values below 5×10^{-7} are removed. This left a total of 422,006 SNPs for type 1 diabetes, 403,301 for arthritis, 405,306 for Crohn's disease, and 402,532 for type 2 diabetes. It was confirmed that the same significant SNPs with the same p-values were reported by the hybrid strategy programs as published in the original WTCCC study (Table 3 of Wellcome Trust Case Control Consortium [97]).

Simulated Data

The GWAsimulator program produces case and control genome-wide SNP genotypes under a logistic regression disease model. It takes phased genotype data as input and simulates SNP genotypes with the same linkage disequilibrium as the input [78]. It outputs data in the encoded numerical format described earlier (i.e., the number of copies of a selected allele). The HapMap CEU phased genotypes provided with the software package were used as input. These genotypes were produced by the Illumina HumanHap300 SNP chip. The program generates one causal SNP on a specified position of a chromosome and then simulates remaining SNPs according to a moving window algorithm [65].

Ten and 15 randomly selected SNPs are randomly specified as causal, one per chromosomes 1 through 15 with relative risks of 1.25, 1.5 and 2. For each setting, 50 genome-wide association studies of disease prevalence .01 and 1,000 case and 1,000 control subjects were simulated. 1,000 SNPs were simulated on either side of each causal one. This adds up to a total of approximately 30,000 SNPs for 15 causal alleles and 20,000 for ten causal alleles for each case and control sample.

All simulated studies, input control files and HapMap CEU phased genotypes to the GWAsimulator program are provided at http://www.cs.njit.edu/usman/SVMSNP.

5.3 Results

First, a comparison of the hybrid strategy to the baseline chi-square and the SVM SNP rankings on simulated data is presented. This is then compared to more methods on four real datasets. Finally, the prediction accuracy of the industry standard disease risk estimator with replicated·SNPs and as a function of top ranked SNPs in the hybrid, chi-square, and SVM rankings on the same four real datasets is examined.

In the simulated datasets the SVM ranking of all SNPs in the study is computed and the loss-complexity tradeoff parameter C is set to $\frac{1}{\sum_i x_i^T x_i}$ where x_i is the encoded SNP genotype vector for the i^{th} subject. This is the default of C computed by the SVM-light software package used in this study. In the real datasets, the SVM was run on just the top 25,000 chi-square ranked SNPs due to running time considerations.

5.3.1 Ranking of Causal SNPs in Simulated Data

The rank of causal SNPs in a given ordering is measured using the *rank-sum*, which is just the sum of the ranks of causal SNPs. For example, a SNP ordering with all 15 causal alleles ranked 1 through 15 would have a rank-sum of $\sum_{i=1}^{15} i = 120$ which is the lowest attainable value for 15 alleles. For ten alleles this value is 55.

The optimal value of r was selected from the set $\{25,50,100\}$ and the optimal value of C was selected from 1 to 10^{-6} in increments of 10^{-1}. For a given dataset, the rank-sum of the hybrid, chi-square, and SVM SNP rankings were computed within the optimal r. Table 5.1 lists the mean rank-sums of the three SNP orderings across the 50 simulated studies. Clearly the hybrid ranks causal alleles better than chi-square and SVM.

Column 5 shows that the hybrid and chi-square differences are significant at all relative risks with 15 causal alleles but only relative risk of 2 with ten causal alleles. Column 6 shows that the hybrid and SVM differences are significant mainly at relative risk of 2 with ten and 15 causal alleles and relative risk of 1.25 with 15 causal alleles. The table shows that the improvement given by the hybrid strategy over chi-square and SVM decreases as one moves to lower relative risks with few causal alleles. The same observations are made in real data below. There it can be seen that the hybrid ranks replicated SNPs higher than chi-square and SVM in type 1 diabetes, arthritis, and Crohn's disease studies and comparable in type 2 diabetes which has relatively much lower signal strength than the first three.

The mean number of causal alleles within the optimal r across the 50 studies is shown in parenthesis in Column 2. Since this is less than the total number of causal alleles this shows that the optimal r is a conservative value.

Table 5.1 Mean Rank-Sum of SNP Orderings Given by the Hybrid (Denoted as Hyb), Chi-square, and the SVM. In Parenthesis in Column 2 is the Mean Number of Causal SNPs Found Within the Optimal Value of r Given by the Hybrid Strategy

Data	χ^2	SVM	Hyb	χ^2 p-values	SVM p-values
2(15)	222(13.6)	190	123	10^{-9}	10^{-7}
1.5(15)	165(11.3)	124	116	10^{-6}	0.33
1.25(15)	48(4.4)	61	33	10^{-3}	10^{-4}
2(10)	93(9.2)	97	50	10^{-8}	10^{-8}
1.5(10)	82(8.3)	74	67	.3	.3
1.25(10)	48(3.5)	47	34	.26	.2

5.3.2 Ranking of Replicated SNPs and Regions on Real Data

In the previous section it was shown that causal SNPs in simulated data are moved to higher ranks by the hybrid. Can the same be said for real data? In real studies the causal SNP may not necessarily be sequenced, but it can be expected that they will be present in future studies as the genome coverage increases and cost of sequencing technology drops. At this time though, the rank of replicated SNPs and known associated regions can be measured as defined by linkage disequilibrium [97].

The ranks of replicated SNPs are examined in four real WTCCC studies of decreasing signal strength: type 1 diabetes, rheumatoid arthritis, Crohn's disease, and type 2 diabetes. The p-values and odds ratios of the most significant SNPs in these studies are 10^{-140} and 2.9, 10^{-75} and 2.0, 10^{-14} and 1.3, and 10^{-12} and 1.26 respectively.

For type 1 diabetes, associated regions are available in a curated table at the Type 1 Diabetes Consortium [63]. For previously replicated SNPs, one can refer to a recent paper [66] that lists such SNPs for the four diseases. Curated associated regions for these diseases are not publicly available at this time.

The hybrid SNP rankings are obtained with optimal values of r from the set $\{25,50,100,250,500,1000\}$ and optimal values of C from 1 through 10^{-8} in increments of 10^{-1}. For Crohn's disease and type 2 diabetes this work also looks at values of C from .5 through 5^{-8} in increments of 10^{-1}. This work also compute the SNP rankings given by the SVM discriminant, SVM-RFE and the HMM. Due to running time considerations, SVM-RFE is run starting from the top 25,000 chi-square ranked SNPs, removing bottom 1,000 SNPs after each iteration, and stopping when 1,000 SNPs remain. The same default value of C is used as for the SVM (described above). This project tried to compute the penalized ML ranking of SNPs, but the program ended after a long computation without any result.

5.3.2.1 Type 1 Diabetes. A comparison of the ranking of previously replicated type 1 diabetes SNPs given by the different methods is given in Table 5.2. The hybrid ranking for the optimal r (500) and twice that value are examined to gain more coverage of associated SNPs. The ranking given by the hybrid strategy at the optimal r places all the SNPs at higher positions than chi-square, SVM, SVM-RFE, and the HMM except for two: rs9272346, which is ranked comparably to other methods, and rs17696736, which is ranked better than all methods but comparable to SVM-RFE.

At twice the optimal r the hybrid strategy ranks all SNPs better than chi-square and HMM except for rs9272346 which is comparable. Compared to SVM it ranks seven out of ten SNPs better, one at the same position, and two worse. SVM-RFE misses two SNPs when it is stopped at 1,000 SNPs. Out of the remaining eight the hybrid ranks six better and one the same.

Table 5.2 Ranking of Different Methods of the WTCCC Type 1 Diabetes Previously Replicated SNPs [66]. The Optimal r Given by the Hybrid is 500

SNP	χ^2 p-value	χ^2	SVM	SVM - RFE	HMM	Hyb r=500	Hyb r=1K
rs9272346	5.40E-134	1	2	2	3	2	2
rs6679677	3.80E-26	96	49	13	113	7	6
rs17696736	1.50E-14	190	77	12	313	15	60
rs2292239	1.80E-09	330	78	323	553	76	66
rs705702	4.80E-07	438	145	725	965	143	225
rs12708716	5.90E-07	449	155	390	790	78	310
rs17388568	2.80E-06	521	113	182	1115	NA	34
rs2542151	1.10E-05	568	326	70	239813	NA	14
rs12251307	1.30E-03	777	2032	NA	27886	NA	86
rs3087243	1.30E-03	783	1004	NA	11767	NA	379

Note that the HMM ranking of replicated SNPs is considerably worse than the other ones. It is possible that the HMM (or any other method for that matter) ranks SNPs in linkage disequilibrium with the replicated ones higher thus still making the detection of a region possible. Fortunately for type 1 diabetes, one can refer to a curated table of known associated regions to determine the ranks of these regions.

A comparison of the number of known associated regions (true positives) and false ones detected by top ranked SNPs in different orderings is given in Table 5.3. This project is interested in the number of known associated regions captured by SNPs and not necessarily the number of SNPs from a given one. For example, a ranking where the top five SNPs represent five different true associated regions is much more useful than one where the top five SNPs are from the same one region. The start and stop positions for these regions are specified by the Type 1 Diabetes Consortium [63].

A given ranking of SNPs is traversed from top to bottom and each known associated region the SNP belongs to is considered as a true positive. During this traversal if an encountered SNP is not from a known associated region then, as a conservative heuristic, a region of 20 SNPs around it (ten on either side) is defined as a false positive region, the false positive counter is incremented, and the traversal proceeds. The region is not considered again (regardless of true or false) if subsequent SNPs from it are encountered during the traversal.

Table 5.3 Number of False and True Positive Regions Identified by Top Ranked SNPs Given by Different Methods. *k* Denotes the Number of Top Ranked SNPs Examined in Each Ranking, FP Denotes the Number of False Positives Regions, and TP Denotes the Number of True Positives (Known Associated Regions). [1]400 is the Number of SNPs Given by the Bonferroni Correction

k	χ^2	SVM	SVM-RFE	HMM	Hyb r=500	Hyb r=1K
25	FP=0	FP=0	FP=2	FP=4	FP=0	FP=5
	TP=1	TP=1	TP=3	TP=1	TP=3	TP=3
50	FP=0	FP=0	FP=23	FP=4	FP=0	FP=21
	TP=1	TP=2	TP=3	TP=1	TP=4	TP=5
100	FP=0	FP=0	FP=56	FP=4	FP=0	FP=44
	TP=2	TP=4	TP=5	TP=1	TP=6	TP=8
200	FP=0	FP=14	FP=131	FP=4	FP=0	FP=87
	TP=3	TP=6	TP=9	TP=1	TP=6	TP=11
400[1]	FP=0	FP=119	FP=294	FP=4	FP=1	FP=121
	TP=6	TP=16	TP=13	TP=3	TP=6	TP=14

In Table 5.3 one can see that the HMM identifies associated regions in the top ranked SNPs even though replicated ones are ranked low (as shown in Table 5.2). It was also found that the top 100 SNPs in the SVM ordering with the optimal value of $r = 500$ detect all associated regions within this value of r. This is a significant advantage over the chi-square and SVM rankings; the former detects only two regions within the top 100 ranked SNPs and the latter only four. This also highlights the main advantage of the hybrid strategy over chi-square and SVM separately. It lifts certain SNPs from associated regions to higher ranks thus making the region detectable by examining a fewer number of top ranked SNPs compared to chi-square and SVM.

The hybrid ranking with $r = 1000$, in comparison to the optimal r, contains more true positives and more false positives. However, compared to SVM-RFE it contains fewer false positives throughout and gains a clear advantage at all values of k (see the Table 5.3 caption for definition of k).

The improvement by the hybrid strategy over chi-square and SVM comes from selecting the correct value of r. Since SVM and SVM-RFE use an arbitrarily large value of r as the starting point the SVM ranking given at that initial value ranks many non-associated SNPs higher than associated ones (as column 3 of Table 5.3 shows). In subsequent iterations this error accumulates which leads to a poor ranking of associated SNPs and regions at the stopping point.

5.3.2.2 Rheumatoid Arthritis, Crohn's disease, and Type 2 Diabetes.

Table 5.4 Chi-square and SVM Ranking of Rheumatoid Arthritis, Crohn's Disease, and Type 2 Diabetes Previously Replicated SNPs [66]

Rheumatoid arthritis						
SNP	χ^2 p-value	χ^2	SVM	SVM-RFE	Hyb r=100	Hyb r=500
rs6457617	4.40E-75	1	1	7	18	3
rs6920220	1.70E-05	242	71	NA	NA	83
rs3890745	4.00E-05	268	57	530	NA	61
rs1678542	1.30E-04	334	474	92	NA	72
Crohn's disease						
SNP	χ^2 p-value	χ^2	SVM	SVM-RFE	Hyb r=50	Hyb r=100
rs3828309	3.90E-13	4	5	232	18	37
rs17234657	2.20E-12	5	13	12	0	0
rs9292777	1.20E-11	9	7	11	9	36
rs17221417	4.40E-11	17	6	281	17	26
rs9858542	3.00E-08	31	848	NA	4	33
rs10883365	5.70E-08	36	17	660	28	49
rs2542151	2.10E-07	51	93	99	NA	6
rs11747270	2.20E-07	54	1357	NA	NA	30
rs6596075	3.30E-06	81	835	NA	NA	57
rs6908425	1.30E-05	96	165	234	NA	5
rs12035082	1.30E-05	98	85	22	NA	15
rs4263839	1.40E-05	99	128	NA	NA	4

Table 5.4 Chi-square and SVM Ranking of Rheumatoid Arthritis, Crohn's Disease, and Type 2 Diabetes Previously Replicated SNPs [66] (Continued)

Type 2 diabetes						
SNP	χ^2 p-value	χ^2	SVM	SVM-RFE	Hyb r=50	Hyb r=100
rs4132670	1.50E-11	2	2	9	24	34
rs8050136	5.40E-08	11	11	211	34	55
rs7961581	2.90E-05	45	30	299	10	8

In Table 5.4, the hybrid, chi-square, SVM, and SVM-RFE ranking of arthritis, Crohn's disease, and type 2 diabetes replicated SNPs are compared in the respective WTCCC studies. For arthritis a small drop in rank is found at the optimal $r = 100$. Note that the optimal r does not cover enough replicated SNPs. This is not surprising since the results on simulated data show that the optimal value is conservative. At five times the optimal r, which provides larger coverage of replicated SNPs, improved ranking of downstream SNPs are found compared to the chi-square ordering. Compared to the SVM ranking SNP rs1678542 is ranked much better by the hybrid. Although the remaining three are lower than SVM the differences are very small. SVM-RFE misses SNP rs6920220 when stopped at a 1000 SNPs and ranks SNP rs3890745 much lower than the hybrid.

For Crohn's disease, the hybrid strategy at the optimal $r = 50$ was found to have better ranks in three out of six SNPs compared to chi-square. Of the remaining three, two are unchanged and one is worse. Compared to SVM the hybrid ranks SNP rs9858542 significantly better but the remaining are comparable. At twice the optimal r the hybrid ranks seven of the twelve better than chi-square and places many other SNPs at significantly higher positions than the SVM. The SVM-RFE ranking misses four SNPs when stopped at 1000 and of the remaining most are ranked lower than the hybrid.

For type 2 diabetes, an improvement was not found over chi-square and SVM with the hybrid strategy. Of the three replicated SNPs covered at the optimal $r = 50$ (and twice that value) the hybrid strategy improves the rank of just one. The performance of the hybrid strategy on the type 2 diabetes study is not too surprising given the results on simulated data. No significant improvement was seen in SNP rankings when the data has few causal alleles of low relative risk.

5.3.3 Risk Prediction Accuracy of Discriminative SNPs

The previous two subsections establish that the hybrid can improve the rank of causal SNPs in simulated data and replicated SNPs on real data with an automatically determined value of C and r, provided that the signal strength is moderate to high. Would the top ranked hybrid SNPs also serve as better predictors of disease risk than previously replicated ones as well as top ranked chi-square and SVM ones? This question is investigated here on the same four real studies examined above.

Accuracy of risk prediction is measured by the area under curve (AUC) of the composite odds ratio score. This score is the industry standard disease risk estimator [82, 87] and has been studied in several previous papers [66, 67, 73, 98]. In these experiments, the HMM and SVM-RFE rankings are excluded. This is because they are not really designed for SNP selection that will optimize risk prediction accuracy and their performance in ranking replicated SNPs in real data (shown above) is comparable to chi-square and SVM.

For each of the four diseases, a random sample of 90% of case and controls was extracted for training and the remainder was used as test. From the training, the hybrid, chi-square and SVM rankings are computed and AUC of the composite odds ratio score of the validation set as a function of top ranked SNPs given by the three methods is measured. This is repeated five times and the mean AUC across the five random training-validation splits is plotted. Figures 5.1 and 5.2 show the mean AUC with previously replicated SNPs and as a function of top ranked hybrid, chi-square, and SVM ranked SNPs on the type 1 diabetes and arthritis studies. The same figures for Crohn's disease and type 2 diabetes are shown in Appendix E.

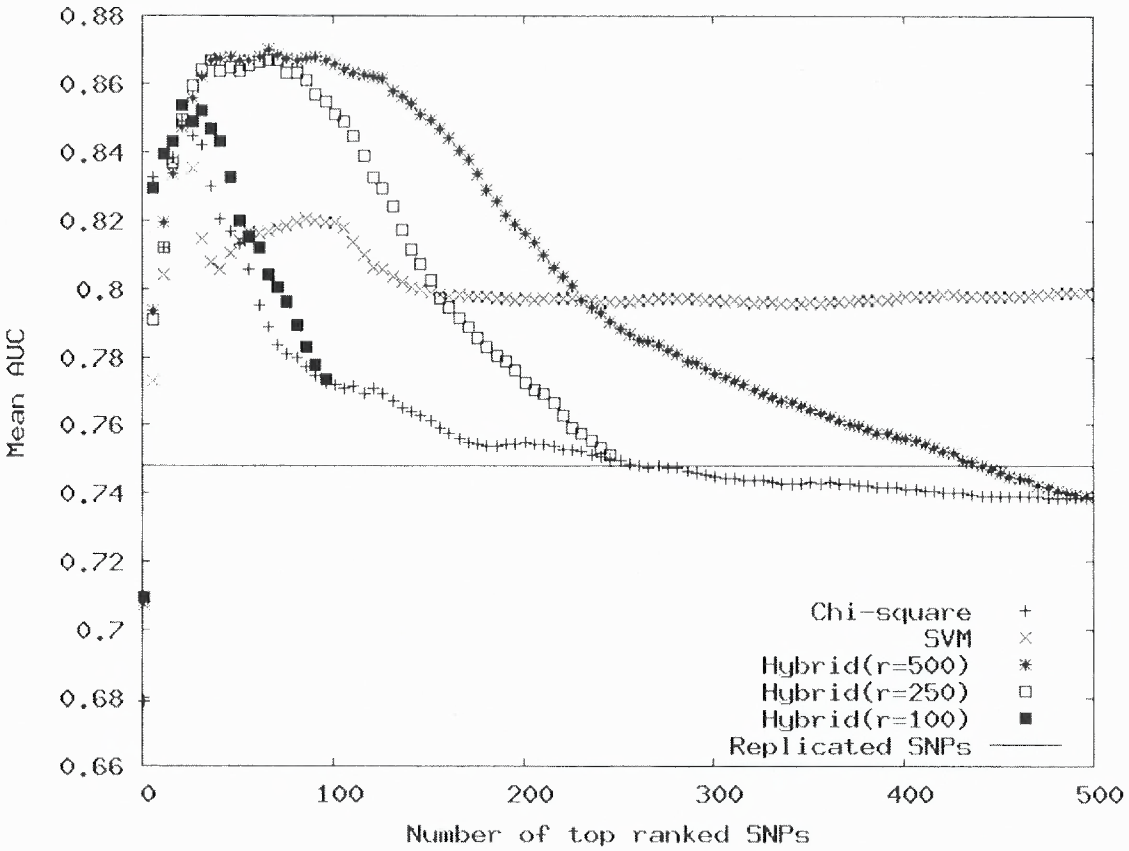


Figure 5.1 AUC of composite odds ratio score on type 1 diabetes study.

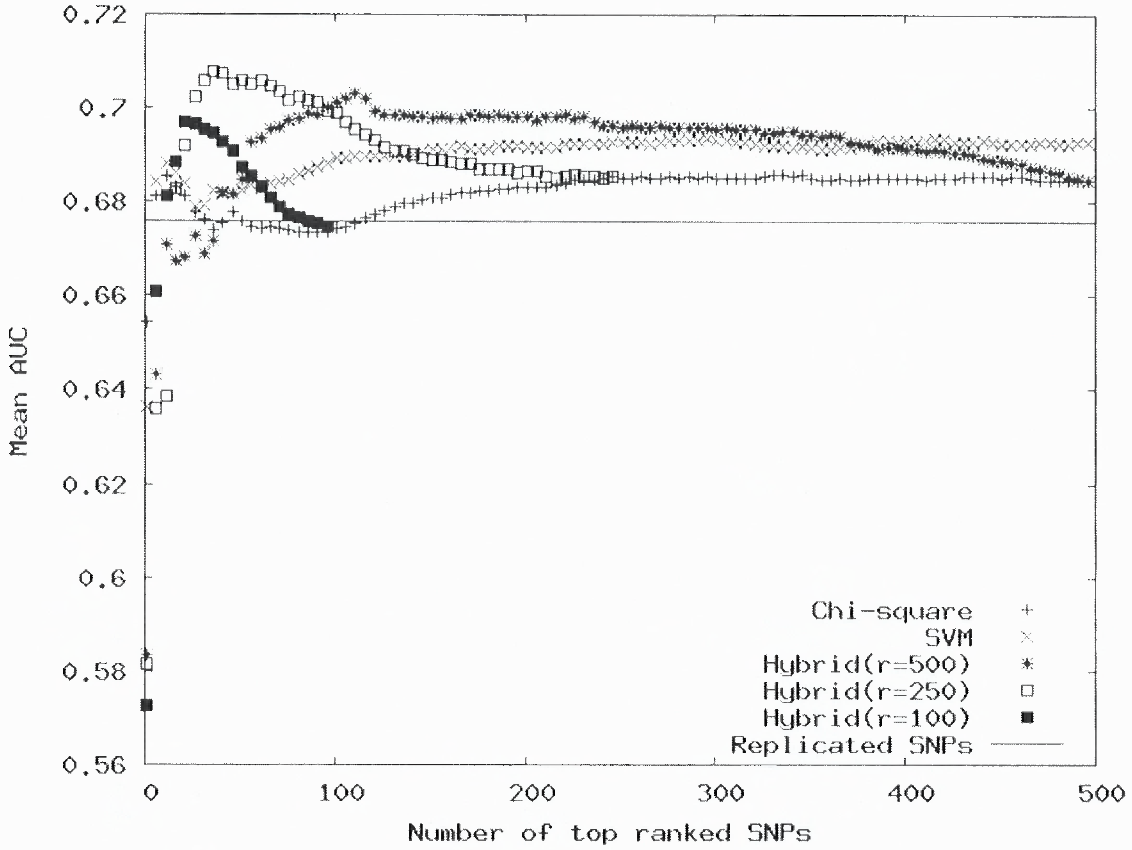


Figure 5.2 AUC of composite odds ratio score on arthritis study.

Figures 5.1 and 5.2 show that top ranked hybrid SNPs achieve a higher AUC than replicated SNPs and top ranked chi-square and SVM SNPs. Across the five runs, different optimal values of r are found. In type 1 diabetes the optimal r is 250 in three of the five training-validation splits. With this r the mean AUC improvement over replicated SNPs and top ranked chi-square and SVM SNPs is 12%, 2% and 2% respectively with an economical set of 37 SNPs (see Table 5.5).

In the arthritis study the optimal r is 250 in four of the five training-validation splits. With this r the improvement over replicated and top ranked chi-square and SVM SNPs is 3%, 2%, and 1% respectively but with much fewer SNPs than top ranked chi-square and SVM ones. In the chi-square and SVM rankings 331 and 881 SNPs are required to achieve their highest accuracies whereas the hybrid strategy requires only 36 (see Table 5.5).

In the Crohn's disease study the optimal r is 100 across all five training-validation splits. However, at this value there is no improvement over replicated SNPs or top ranked chi-square and SVM ones. At a large value of $r = 500$ an improvement of 1% is found, but with many SNPs. In the type 2 diabetes the optimal r is 50 in two of the five training-validation splits, 250 for other two, and 100 for one. No improvement in AUC with SVM SNPs was seen for these settings. In fact previously replicated SNPs have the highest AUC in this study.

Table 5.5 The Highest Accuracy (HA) and Number of SNPs Required to Achieve this in Chi-square and SVM Rankings and with Replicated SNPs (Denoted as Rep. SNPs Below)

Method	T1D		RA		CD		T2D	
	HA	SNPs	HA	SNPs	HA	SNPs	HA	SNPs
Rep SNPs	7.50E-01	19	0.68	8	0.65	28	0.6	15
χ^2	8.50E-01	22	0.69	331	0.65	321	0.59	65
SVM	8.50E-01	22	0.7	881	0.65	265	0.59	875
Hyb(r=500)	8.70E-01	66	0.7	111	0.66	337	0.57	489
Hyb(r=250)	8.70E-01	37	0.71	36	0.65	171	0.58	175
Hyb(r=100)	8.50E-01	23	0.7	21	0.64	83	0.59	99
Hyb(r=50)	8.60E-01	6	0.69	11	0.64	31	0.58	39

5.4 Discussion

The main contribution in this work is a hybrid strategy that combines the SVM and the chi-square statistic to produce a ranking of SNPs such that causal and replicated ones are highly placed. Without the hybrid it is shown that the SVM if applied directly to a genome-wide study (or applied within the SVM-RFE framework) does not necessarily rank causal or known replicated SNPs at high positions. As mentioned earlier, the SVM component of the hybrid can be replaced with a different regularized risk minimizer but it's not clear if the results presented here would still hold.

The motivation for the hybrid strategy is best described by Table 5.3. There it can be seen that the chi-square statistic ranks many SNPs from known associated regions at high positions with hardly any false positives. The SVM, on the other hand, has SNPs from more associated regions ranked high but also many SNPs from false positive regions. The hybrid achieves a balance between them. In just the top 100 ranked SNPs it detects all true positives in the optimal r with zero false positives.

The top ranked SNPs produced by the hybrid lead to better risk prediction accuracy on studies with moderate to high signal even though the area under curve is measured by the composite odds ratio score which is not part of the hybrid strategy. The hybrid uses a very simple nearest centroid classifier to determine the discriminative strength of a set of SNPs. It is interesting to observe that the hybrid strategy also ranks replicated SNPs high even though it is geared towards finding discriminative SNPs.

The benefits of the hybrid strategy are currently limited to studies with moderate to high signal strength. It may be hard for any other method to rank replicated SNPs in low signal studies higher than the basic chi-square or SVM. It can be seen here that even with replicated SNPs for type 2 diabetes and and Crohn's disease the mean AUC is .6 and .65. Both are too low to be of much use in practice as discussed in recent work [67, 68, 77]. However, as more data is collected and deeper sequencing is performed (such as whole genome coverage) one may find variants with moderate to high signals. On such studies it is expected that the hybrid strategy will be more useful.

The hybrid strategy could in principle be applied to the gene selection problem studied in the original SVM-RFE paper [69]. Since it is geared towards finding parameters that best discriminate the two classes it may obtain a small set of genes that obtain high classification accuracy for that problem (as demonstrated for type 1 diabetes and arthritis here). The hybrid strategy can also be applied to detect interacting SNPs as follows: compute the chi-square ranking of SNPs, select a liberal number of top ranked SNPs (say 500), compute all possible pairs of these SNPs [84], re-rank the pairs with chi-square, and then apply the hybrid to the chi-square ranked paired dataset. Finally, a finer resolution of values of t is likely to produce better rankings but will naturally increase the running time.

5.5 Conclusion

The experimental results presented here lead one to conclude that the hybrid strategy provides an ordering of SNPs where top ranked ones simultaneously contain more causal and replicated SNPs and predict disease risk better than previously replicated ones and top ranked chi-square and SVM ones except for studies with very low signal strength. In type 1 diabetes, this report also finds a larger coverage of known associated regions (without the expense of false positives) in top ranked hybrid SNPs compared to other methods.

APPENDIX A

SUMMARY OF SIMULATION RESULTS FOR PHYLOGENY RECONSTRUCTION

Table A.1 describes the results of the comparative study of phylogeny reconstruction programs using simulation.

Table A.1 Summary of Simulation Results

Scale / Deviation	CLUSTALW	MUSCLE	MUSCLE -PROG	MUSCLE -UPGMA	MUSCLE PARS1	MUSCLE PARS2	Best Diff[2]
							.
Percent error rates for 100 taxa, 500 sequence length, indel probability 5x10^{-5}							
16 / 2	9.4	9.1	9.5	9.4	8.7 (0.1)	8.8	0.4
32 / 2	8.7	8.7	8.6	8.7	8.2 (0.3)	8.5	0.4
64 / 2	10.2	10.1	10.9	10.3	11.0	9.9 (0.2)	0.2
16 / 4	13.8	13.8	13.9	13.7	13.5 (tie)	13.5 (tie)	0.2
32 / 4	13.3	13.0 (tie)	13.1	13.2	13.0 (tie)	13.0 (tie)	---
64 / 4	14.1	14.3	15.0	14.0 (0.1)	15.5	14.7	-0.7
Percent error rates for 100 taxa, 1000 sequence length, indel probability 5x10^{-5}							
16 / 2	6.1	6.0	5.9	5.8	5.4 (tie)	5.4 (tie)	0.4
32 / 2	4.8	5.0	4.9	4.8	4.6 (0.1)	4.7	0.2
64 / 2	7.0	6.9	7.1	6.5	6.4 (0.1)	6.9	0.1
16 / 4	9.1	9.1	8.9	9.0	8.9	8.8 (0.1)	0.1
32 / 4	8.9	8.5	8.7	8.4	8.0 (0.1)	8.1	0.4
64 / 4	13.4	11.5 (0.7)	13.8	12.2	14.0	12.5	-1.0
Percent error rates for 100 taxa, 500 sequence length, indel probability 5x10^{-4}							
16 / 2	10.9	10.9	10.8	10.7	10.3 (tie)	10.3 (tie)	0.4
32 / 2	11.5	10.2	9.0(0.2)	9.2	9.3	9.2	-0.2
64 / 2	16.6	25.3	19.7	17.6	17.3	16.4(0.2)	0.2
16 / 4	13.9	13.9	13.7	13.5	12.7(0.3)	13.0	0.8
32 / 4	17.5	16.4	14.9	14.6	14.4	13.7(0.7)	0.9
64 / 4	24.4	30.6	24.2	23.3	22.9	22.6(0.3)	0.7
Percent error rates for 100 taxa, 1000 sequence length, indel probability 5x10^{-4}							
16 / 2	5.7	5.6	5.6	5.8	5.2 (0.3)	5.5	0.4
32 / 2	6.9	6.7	6.1	6.4	6.0	5.9 (0.1)	0.2
64 / 2	13.9 (0.4)	18.3	18.5	15.9	15.9	14.3	-0.4
16 / 4	8.8	9.0	8.5	8.5	8.3 (tie)	8.3 (tie)	0.2
32 / 4	13.4	12.3	10.6	10.5 (tie)	11.0	10.5 (tie)	---
64 / 4	23.4	26.7	23.7	21.7	23.1	20.6(1.1)	1.1

Table A.1 Summary Of Simulation Results (Continued)

Scale / Deviation	CLUSTALW	MUSCLE	MUSCLE -PROG	MUSCLE -UPGMA	MUSCLE PARS1	MUSCLE PARS2	Best Diff[2]
Percent error rates for 200 taxa, 500 sequence length, indel probability 5x10^{-5}							
16 / 2	11.1	11.2	11.2	11.3	10.5(0.7)	10.5(0.7)	0.7
32 / 2	8.3	8.2	7.9(tie)	7.9 (tie)	7.9 (tie)	8.0	---
64 / 2	10.2	11.2	11.4	9.6 (0.4)	11.3	10.0	-0.4
16 / 4	15.3	15.5	15.4	15.5	13.8(0.2)	14.0	1.5
32 / 4	11.5	11.3	11.4	11.4	11.3	11.2 (0.1)	0.1
64 / 4	17.0	16.5	17.3	15.4 (0.3)	17.0	15.7	-0.3
Percent error rates for 200 taxa, 1000 sequence length, indel probability 5x10^{-5}							
16 / 2	6.2	6.3	6.3	6.3	5.7(tie)	5.7 (tie)	0.6
32 / 2	5.6	5.6	5.5	5.5	5.4(tie)	5.4 (tie)	0.1
64 / 2	7.2	7.7	8.2	6.9 (tie)	8.2	6.9 (tie)	---
16 / 4	9.4 (tie)	9.5	9.4 (tie)	9.4 (tie)	9.5	9.5	-0.1
32 / 4	9.0	8.9	8.8 (tie)	8.8 (tie)	8.9	8.8 (tie)	---
64 / 4	14.4	13.6	14.4	12.8	14.2	12.7 (0.1)	0.1
Percent error rates for 200 taxa, 500 sequence length, indel probability 5x10^{-4}							
16 / 2	11.9	11.7	11.2	11.2	10.2	9.7 (0.5)	1.5
32 / 2	12.5	14.7	10.3	10.0	10.0	9.5 (0.5)	0.5
64 / 2	19.0 (0.4)	37.4	22.2	20.7	19.9	19.4	-0.4
16 / 4	16.1	16.4	15.3	15.3	14.4	14.2 (0.2)	1.1
32 / 4	17.0	19.6	15.6	15.4	14.6	14.5 (0.1)	0.9
64 / 4	26.6	44.0	26.6	26.1	25.6	23.9 (1.7)	2.2
Percent error rates for 200 taxa, 1000 sequence length, indel probability 5x10^{-4}							
16 / 2	7.2	7.6	7.1	6.9	6.6	6.4 (0.2)	0.5
32 / 2	9.5	10.4	6.8	6.8	6.6 (tie)	6.6 (tie)	0.2
64 / 2	15.8 (0.9)	28.4	19.8	18.4	17.6	16.7	-0.9
16 / 4	11.1	11.4	10.1	10.1	9.5 (0.2)	9.7	0.6
32 / 4	14.4	16.0	11.9	11.8	11.2 (0.1)	11.3	0.6
64 / 4	23.7	36.0	24.6	22.9	22.6	21.5 (1.1)	1.4
Percent error rates for 400 taxa, 500 sequence length, indel probability 5x10^{-5}							
16/2	12.6	12.6	12.6	12.6	11.5 (0.1)	11.6	1.1
32/2	8.7	8.6	8.6	8.6	8.3	8.1 (0.2)	0.5
64/2	9.0	10.1	9.6	8.6	9.0	8.3 (0.3)	0.3
16/4	17.8	17.9	17.9	17.9	16.2 (0.2)	16.4	1.6
32/4	13.3	13.3	13.2	13.2	12.8 (0.1)	12.9	0.4
64/4	15.1	15.7	14.7	13.9	14.5	13.5 (0.4)	0.4
Percent error rates for 400 taxa, 1000 sequence length, indel probability 5x10^{-5}							
16 / 2	7.4	7.3	7.4	7.3	7.0 (tie)	7.0 (tie)	0.3
32 / 2	5.5 (tie)	5.6	5.5 (tie)	5.5 (tie)	5.5 (tie)	5.5 (tie)	---
64 / 2	6.5	7.1	6.8	6.0 (0.1)	6.4	6.1	-0.1
16 / 4	10.3	10.3	10.3	10.3	9.6 (tie)	9.6 (tie)	0.7
32 / 4	8.8	8.9	8.5 (tie)	8.7	8.5 (tie)	8.5 (tie)	---
64 / 4	12.2	11.9	11.5	10.9 (0.1)	11.9	11.0	-0.1

Table A.1 Summary Of Simulation Results (Continued)

Scale / Deviation	CLUSTALW	MUSCLE	MUSCLE -PROG	MUSCLE -UPGMA	MUSCLE PARS1	MUSCLE PARS2	Best Diff[2]
Percent error rates for 400 taxa, 500 sequence length, indel probability 5x10^{-4}							
16 / 2	13.1	14.5	12.8	12.7	12.0 (tie)	12.0 (tie)	0.7
32 / 2	11.8	16.3	10.0	9.8	9.4	9.3 (0.1)	0.5
64 / 2	15.9	40.1	17.9	16.6	15.5	15.3 (0.2)	0.6
16 / 4	18.2	19.7	17.6	17.6	15.9 (tie)	15.9 (tie)	1.7
32 / 4	17.0	21.2	15.4	15.6	14.5 (0.1)	14.6	0.9
64 / 4	22.8	44.5	22.9	21.8	22.4	21.5 (0.3)	0.3
Percent error rates for 400 taxa, 1000 sequence length, indel probability 5x10^{-4}							
16 / 2	8.0	9.3	7.6	7.6	7.2 (tie)	7.2 (tie)	0.4
32 / 2	8.2	10.0	6.6	6.4	6.2 (0.1)	6.3	0.2
64 / 2	12.3 (0.6)	33.3	15.0	14.5	13.5	12.9	-0.6
16 / 4	11.6	13.4	11.0	11.0	10.3 (0.1)	10.4	0.7
32 / 4	12.9	15.1	10.6	10.6	10.2 (0.1)	10.3	0.4
64 / 4	20.4	39.5	19.8	19.0	18.6	18.2 (0.4)	0.8
Overall results: number of times each method was best (ties are counted in each occurrence)							
Dev. = 2	5	0	3	5	**16**	**20**	---
Dev. = 4	1	2	3	6	**16**	**21**	---
Total	6	2	6	11	**32**	**41**	---

Note:

[1] Best scoring alignments (across all six possibilities) also included the percent difference between it and the next best scoring alignment (again, across all six possibilities) in parentheses.

[2] In the final column, the difference between the best scoring MUSCLE-PARS alignment and the best of the remaining four alignments is presented.

APPENDIX B

PROBALIGN RNA-GENOME BENCHMARK STATISTICS

Table B.1 below lists some characteristics of the 22 RNA families in the RNA-Genome benchmark.

Table B.1 Statistics for All 22 RFAM RNA Families Used in the Study

RFAM RNA family	Average pairwise sequence identity	Sequence length standard deviation	Number of sequences in seed family alignment	Number of pairwise alignments in benchmark
5S_rRNA	55	2.58	50	49
U1	56	6.67	50	141
Trna	39	4.9	50	342
RNaseP_bact_a	59	37.78	50	143
RNaseP_bact_b	59	37.84	50	23
U3	45	55.94	21	20
U4	56	11.04	26	69
SRP_euk_arch	45	10.45	50	331
tmRNA	40	31.51	50	342
Intron_gpI	43	77.46	30	71
SECIS	41	3.16	50	347
IRE	54	1.43	39	231
THI	55	17.99	50	347
Hammerhead_1	56	31.95	50	49
Purine	50	0.85	12	59
Lysine	45	8.47	19	147
SRP_bact	50	9.19	42	348
SSU_rRNA_5	48	128.30	50	97
T-box	51	2.49	14	62
glmS	50	26.90	6	19
RNaseP_arch	51	67.61	34	156
IRES_Cripavirus	49	4.92	7	36

Note: First, subsets of each RFAM seed family alignment containing a maximum of 50 randomly selected sequences. For each subset, directions listed in the main report were followed to construct the benchmark.

APPENDIX C

COMMAND LINE PARAMETERS FOR PROGRAMS USED IN GENERATING THE RNA-GENOME BENCHMARK

In the descriptions below <data> refers to unaligned query and genome sequence in FASTA format and <query> and <genome> refer to the separate sequences also in FASTA format.

```
Probalign:   probalign –nuc –T 7 –go 32 –ge 2 <data>

SSEARCH:     ssearch –H –q –d 1 –a  –f 10 –e 4  -O ssearch.out  <query> <genome>

BLAST:       bl2seq –p blastn –G 8 –E 6 –W 4 –S 1 –r 5 –q -4 –i <query> -j <genome>

ClustalW:    clustalw –infile=<data> -outorder=input –output=fasta –outfile=cw.out

HMMER:       (1) hmmbuild –nucleic –informat=PHYLIP –f –F model.hmm <query>
             (2) hmmsearch model.hmm <genome>
```

APPENDIX D

DESCRIPTION OF PROBALIGN

The sections below explain: the maximal expected accuracy alignment methodology, how match or posterior probabilities are used, and how to compute these probabilities using partition function matrices. As explained below, posterior probabilities can be tied with expected accuracy alignment in the Probalign program.

Posterior probabilities and maximal expected accuracy alignment

Most alignment programs compute an optimal sum-of-pairs alignment or a maximum probability alignment using the Viterbi algorithm (Durbin *et al.* [105]). An alternative approach is to search for the maximum expected accuracy alignment [9, 105]. The expected accuracy of an alignment is based upon the posterior probabilities of aligning residues in two sequences.

Consider sequences x and y and let a^* be their true alignment. Following the description in Do *et al.* [9], the posterior probability of residue x_i aligned to y_j in a^* is defined as

$$P(x_i \sim y_j \in a^* \mid x,y) = \sum_{a \in A} P(a \mid x,y) 1\{x_i \sim y_j \in a\} \qquad \text{(D.1)}$$

Where, A is the set of all alignments of x and y and *1(expr)* is the indicator function which returns 1 if the expression *expr* evaluates to true and 0 otherwise. $P(a|x,y)$ represents the probability that alignment a is the true alignment a^*. From hereon, this dissertation represents the posterior probability as $P(x_i \sim y_j)$ with the understanding that it represents the probability of x_i aligned to y_j in the true alignment a^*.

Given the posterior probability matrix $P(x_i \sim y_j)$, one can compute the maximal expected accuracy alignment using the following recursion described in Durbin *et al.* [105].

$$A(i,j) = \max \begin{Bmatrix} A(i-1,j-1) + P(x_i \sim y_j) \\ A(i-1,j) \\ A(i,j-1) \end{Bmatrix} \tag{D.2}$$

According to equation (D.1) as long as there is an ensemble of alignments A with their probabilities $P(a|,x,y)$ one can compute the posterior probability $P(x_i \sim y_j)$ by summing up the probabilities of alignments where x_i is paired with y_j . One way to generate an ensemble of such alignments is to use the partition function methodology, which is described below.

Posterior Probabilities by Partition Function

Amino acid scoring matrices, normally used for sequence alignment, are represented as log-odds scoring matrices (as defined by Dayhoff *et al.* [106]). The commonly used sum-of-pairs score of an alignment a [105] is defined as the sum of residue-residue pairs and residue-gap pairs under an affine penalty scheme.

$$S(a) = T \sum_{(i,j)\in a} \ln(M_{ij} / f_i f_j) + (gap\_penalties) \tag{D.3}$$

Here T is a constant (depending upon the scoring matrix), M_{ij} is the mutation probability of residue i changing to j and f_i and f_j are background frequencies of residues i and j. In fact, it can be shown that any scoring matrix corresponds to a log odds matrix [107, 108].

Miyazawa [46] proposed that the probability of alignment a, $P(a)$, of sequences x and y can be defined as

$$P(a) \propto e^{S(a)/T} \tag{D.4}$$

where, $S(a)$ is the score of the alignment under the given scoring matrix. In this setting one can then treat the alignment score as negative energy and T as the thermodynamic temperature, similar to what is done in statistical mechanics. Analogous to the statistical mechanical framework, Miyazawa [46] defined the partition function of alignments as

$$Z(T) = \sum_{a \in A} e^{S(a)/T} \tag{D.5}$$

where, A is the set of all alignments of x and y. With the partition function in hand, the probability of an alignment a can now be defined as

$$P(a,T) = e^{S(a)/T} \,/\, Z(T) \tag{D.6}$$

As T approaches infinity all alignments are equally probable, whereas at small values of T, only the nearly optimal alignments have the highest probabilities. Thus, the temperature parameter T can be interpreted as a measure of deviation from the optimal alignment.

The alignment partition function can be computed using recursions similar to the Needleman-Wunsch dynamic algorithm. Let Z^{M}_{ij} represent the partition function of all alignments of $x_{1..i}$ and $y_{1..j}$ ending in x_i paired with y_j, and $S_{ij}(a)$ represent the score of alignment a of $x_{1..i}$ and $y_{1..j}$. According to equation (D.5)

$$Z^{M}_{i,j} = \sum_{a \in A_{ij}} e^{S_{ij}(a)/T} = \left(\sum_{a \in A_{i-1\,j-1}} e^{S_{i-1,i-1}(a)/T} \right) e^{s(x_i,y_j)/T} \tag{D.7}$$

where, A_{ij} is the set of all alignments of $x_{1..i}$ and $y_{1..j}$, and $s(x_i,y_j)$ is the score of aligning residue x_i with y_j. The summation in the bracket on the right hand side of equation (D.7) is precisely the partition function of all alignments of $x_{1..i-1}$ and $y_{1..j-1}$. One can thus compute the partition function matrices using standard dynamic programming.

$$\begin{aligned}
Z^{M}_{i,j} &= (Z^{M}_{i-1,j-1} + Z^{E}_{i-1.j-1} + Z^{F}_{i-1,j-1})e^{s(x_i,y_j)/T} \\
Z^{E}_{i,j} &= Z^{M}_{i,j-1}e^{g/T} + Z^{E}_{i.j-1}e^{ext/T} \\
Z^{F}_{i,j} &= Z^{M}_{i-1,j}e^{g/T} + Z^{F}_{i-1.j}e^{ext/T} \\
Z_{i,j} &= Z^{M}_{i,j} + Z^{E}_{i,j} + Z^{F}_{i,j}
\end{aligned} \tag{D.8}$$

Here $s(x,y)$ represents the score of aligning residue x_i with y_j, g is the gap open penalty, and ext is the gap extension penalty. The matrix Z^{M}_{ij} represents the partition function of all alignments ending in x_i paired with y_j. Similarly, Z^{E}_{ij} represents the partition function of all alignments in which y_j is aligned to a gap and Z^F_{ij} all alignments in which x_i is aligned to a gap. Boundary conditions and further details can be obtained from Miyazawa [46].

Once the partition function is constructed, the posterior probability of x_i aligned to y_j can be computed as

$$P(x_i \sim y_j) = \frac{Z^M_{i-1,j-1} Z'^M_{i+1,j+1}}{Z} e^{s(x_i, y_j)/T} \tag{D.9}$$

where, Z'^M_{ij} is the partition function of alignments of subsequences $x_{i..m}$ and $y_{j..n}$ beginning with x_i paired with y_j and m and n are lengths of x and y respectively. This can be computed using standard backward recursion formulas as described in Durbin *et al.* [105].

In equation (D.9) $Z^M_{i-1,j-1}/Z$ and $Z'^M_{i+1,j+1}/Z$ represent the probabilities of all feasible suboptimal alignments (determined by the T parameter) of $x_{1..i-1}$ and $y_{1..j-1}$, and $x_{i+1.m}$ and $y_{j+1..n}$ respectively, where m and n are lengths of x and y respectively. Thus, equation (D.9) weighs alignments according to their partition function probabilities and estimates $P(x_i \sim y_j)$ as the sum of probabilities of all alignments where x_i is paired with y_j.

Maximal Expected Accuracy Alignment using Partition Function Posterior Probabilities

Recall the maximum expected accuracy alignment formulation described earlier. In order to compute such an alignment one needs an estimate of the posterior probabilities. In this report, the partition function posterior probability estimates are utilized for constructing multiple alignments. For each pair of sequences (x, y) in the input, the posterior probability matrix $P(x_i \sim y_j)$ is computed using equation (D.9). These probabilities are subsequently used to compute a maximal expected multiple sequence alignment using the Probcons methodology. First, the probabilistic consistency transformation (described in detail in Do *et al.* [9]) is applied to improve the estimate of the probabilities. Briefly, the probabilistic consistency transformation is to re-estimate the posterior probabilities based upon three-sequence alignments instead of pairwise. Note that this does not mean alignments are recomputed; this estimation (as done in Probcons) is still fundamentally based upon pairwise alignments.

After the probabilistic consistency transformation, sequence profiles are next aligned in a post-order walk along a UPGMA guide-tree. As is commonly done, UPGMA guide trees are computed using pairwise expected accuracy alignment scores. Finally, iterative refinement is performed to improve the alignment. This standard alignment procedure is described in more detail in Do *et al.* [9] and is implemented in the Probcons package (by the same authors).

The Probalign approach is implemented by modifying the underlying Probcons program to read in arbitrary posterior probabilities for each pair of sequences in the input. All use of HMMs in the modified Probcons code is disabled. The probA program of Muckstein *et al.* [47] was modified for computing partition function posterior probability estimates. The Probalign program is represented algorithmically in Figure D.1. The current implementation is a beta version and mainly for proof of concept; however, the open source code is fully functional and is available with full support from http://www.cs.njit.edu/usman/probalign.

Probalign algorithm:

1. For each pair of sequences (x,y) in the input set
    a. Compute partition function matrices $Z(T)$
    b. Estimate posterior probability matrix $P(x_i \sim y_j)$ for (x,y) using equation D.9
2. Perform the probabilistic consistency transformation and compute a maximal expected accuracy multiple alignment: align sequence profiles along a guide-tree and follow by iterative refinement (Do *et. al.*).

Figure D.1 Probalign algorithmic description.

APPENDIX E

SUPPLEMENTARY MATERIAL FOR "A HYBRID STRATEGY FOR RANKING SNPS IN GENOME-WIDE ASSOCIATION STUDIES"

The following sections provide background information regarding encoding of SNP genotype data, chi-square statistic, support vector machine, and composite odds ratio for estimating disease risk.

Numerical Encoding of SNP Genotypes

Before applying chi-square or SVM to SNP genotype data, they need to be converted it to a numerical format by a standard encoding used in population structure identification [89]. Suppose one is given m SNP genotypes $g_i = \{g_{i1}, g_{i2}, \ldots, g_{im}\}$ for each of $i = 1..n$ individuals and m SNP identifiers $s_1, s_2, \ldots, s_m$. Each genotype is of the form $g_{ij} \in \{AA, AB, BB\}$ where A and B are nucleotides (alleles) and are assumed to be alphabetically ordered $(A < B)$. In this work case and control are represented by $y_i \in \{+1, -1\}$ for $i = 1..n$. If $y_i = 1$ then x_i is a case subject and otherwise it is a control. Each genotype g_{ij} is encoded into an integer, thus forming the data matrix M. If $g_{ij} = AA$, M_{ij} is set to 0, else if $g_{ij} = AB$, M_{ij} is set to 1, and otherwise it is set to 2. The encoding used in this research work is the number of copies of the allele with the larger nucleotide. Each row of M represents the genotype of an individual and each column represents a SNP.

Chi-square Statistic

The chi-square statistic has also been referred to as genotypic 2 degree-of-freedom test [97]. The following briefly explain its basics. Define six random variables each of which is binomially distributed $X_i \sim B(n,p_i)$ where n is the total number of subjects and p_i is the probability of success for X_i. Each of these corresponds to the number of case or control subjects with 0, 1, or 2 copies of the allele of interest (see Table E.1). The expected value of each X_i is given by $E(X_i) = np_i$. It can then be shown that the statistic below follows the chi-square distribution with 2 degrees of freedom [74]. This is called the chi-square statistic.

$$\chi^2 = \sum_{i=1}^{6} \frac{(c_i - e_i)^2}{e_i} \tag{E.1}$$

Table E.1 A 2×3 Contingency Table for a Given SNP. Each Entry Denotes Counts of Genotype in Case and Controls. In Parenthesis are Random Variables

	0	1	2
Case	c_1 (X_1)	c_2 (X_2)	c_3 (X_3)
Control	c_4 (X_4)	c_5 (X_5)	c_6 (X_6)

To apply this statistic for detecting SNPs from associated regions let the disease type be given by the random variable D and genotype by G. If it is assumed that these are independent, then $P(D,G) = P(D)P(G)$. These are easy to calculate from counts in the contingency table. For example, $P(G=0) = \frac{(c_1 + c_4)}{n}$ and $P(D = case) = \frac{c_1 + c_2 + c_3}{n}$.

Similarly the expected values of each X_i can be calculated under the null hypothesis and consequently the chi-square statistic. For example, for X_1 its expected value is given by

$$E(X_1) = P(X = case)P(Y = 0)n = \frac{(c_1 + c_2 + c_3)(c_1 + c_4)}{n}$$ under the null hypothesis.

The corresponding p-values can be obtained by referring to the chi-square distribution with 2 degrees of freedom. SNPs with the least p-values deviate from the independence assumption and therefore are of interest.

Support Vector Machine

The support vector machine (SVM) is a powerful discriminative classification algorithm. It makes no assumptions about the underlying (unknown) probability distribution from which the data is drawn. The basic support vector machine algorithm is outlined here and readers can refer to Scholkopf et al. [92] for additional details.

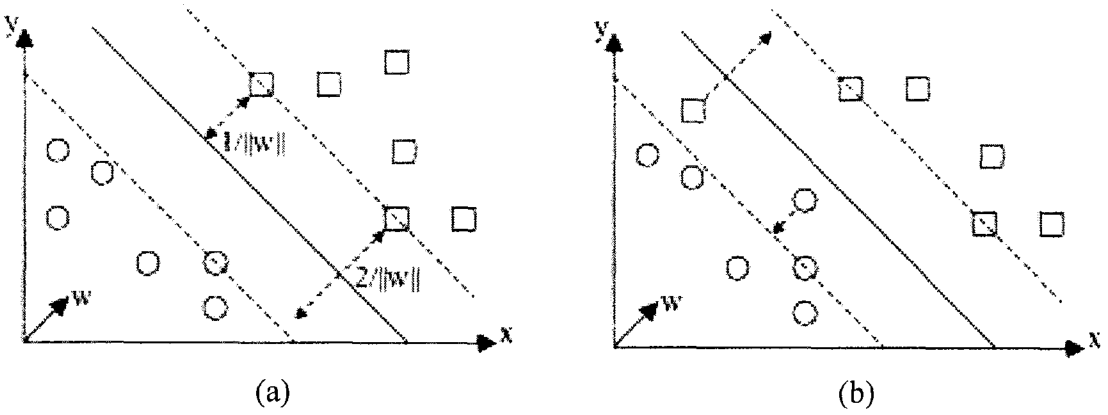


Figure E.1 Toy example of an optimal hyperplane separating points on a plane (illustrated by squares and circles). In (a) $\frac{2}{\| w \|}$ denotes the *margin* of the classifier. Points on the margin are at a distance of $\frac{1}{\| w \|}$. Maximization of the margin can be thought of as minimizing the complexity of the classifier. The example in (b) shows one square misclassified and one circle inside the margin. This is the case when no hyperplane can separate the data points and therefore some points will be necessarily misclassified.

Suppose one is given n vectors $x_i \in R^d$ each with labels $y_i \in \{+1,-1\}$ drawn from a joint probability distribution $P(x,y)$. Referring to Figure E.1(a) suppose the circles represent vectors with labels +1 and the squares represent those with labels -1. The optimally separating hyperplane between these two sets of points is the support vector machine (see Figure E.1). It is defined by a vector $w \in R^d$ and a number w_0. This can be found by solving the following problem with Lagrange multipliers and KKT conditions.

$$\arg\min_{w,w_0} \frac{1}{2} \| w \|^2 + C\sum_i \xi_i$$
$$\text{subject to } y_i(w^T x_i + w_0) \geq 1 - \xi_i \text{ and } \xi_i \geq 0$$

The term $\| w \|^2$ captures the complexity of the classifier and $\sum_i \xi_i$ is the total error on training data. The parameter C controls tradeoff between minimizing complexity and error.

An attractive feature of the SVM classifier is that the probability of misclassifying points drawn from *any* distribution $P(x,y)$ can be bounded by the number of misclassified points available in advance from $P(x,y)$ (also known as training data) plus a term that quantifies the complexity of the classifier [92, 95]. There may be several hyperplanes that separate the two classes with zero misclassifications on the training data. But the optimal one has been shown to minimize classifier complexity as well.

Composite Odds Ratio for Estimating Disease Risk

A standard assumption in disease models is that the the probability of disease given i copies of a risk allele is given by $P(D \mid g_i) = \frac{1}{1+e^{-\alpha+i\beta}}$ for $i = 0,1,2$ [74, 98]. This follows naturally by assuming that the log likelihood ratio is linear and is also known as the logistic regression model [62]. Under this assumption one can estimate α and β by maximum likelihood using a simple gradient descent procedure [62]. This usually converges within a few iterations.

After estimating α and β, e^{β} is used as the odds ratio for the given SNP. This follows naturally by noting that $P(D \mid g_i) = \frac{1}{1+e^{-\alpha+i\beta}}$ can be rewritten as $\ln(\frac{P(D \mid g_i)}{1-P(D \mid g_i)}) = \alpha + i\beta$. The odds ratio $\frac{\frac{Pr(D \mid g_1)}{1-Pr(D \mid g_1)}}{\frac{Pr(D \mid g_0)}{1-Pr(D \mid g_0)}}$ is then given by $\lambda = e^{\beta}$. For two copies of the risk allele one obtains $e^{2\beta} = (e^{\beta})^2 = \lambda^2$. The odds ratio calculated in this manner (under the logistic regression model) does not suffer from bias and stratification problems under simpler models [74].

In this report, it is assumed that each SNP is acting independently. Then the composite odds ratio for several SNPs is defined as $\Pi_i \lambda_i$ where λ_i is the odds ratio of SNP i.

REFERENCES

1. Jones NC, Pevzner P: An introduction to bioinformatics algorithms. Cambridge, MA: MIT Press; 2004.

2. Gusfield D: Algorithms on strings, trees, and sequences : computer science and computational biology. Cambridge England ; New York: Cambridge University Press; 1997.

3. Setubal JC, Meidanis J: Introduction to computational molecular biology. Boston: PWS Pub.; 1997.

4. Edgar RC: MUSCLE: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics* 2004, 5:113.

5. Thompson JD, Plewniak F, Poch O: BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs. *Bioinformatics* 1999, 15(1):87-88.

6. Van Walle I, Lasters I, Wyns L: SABmark--a benchmark for sequence alignment that covers the entire known fold space. *Bioinformatics* 2005, 21(7):1267-1268.

7. Thompson JD, Higgins DG, Gibson TJ: ClustalW: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research* 1994, 27:2682-2690.

8. Katoh K, Kuma K-i, Toh H, Miyata T: MAFFT version 5: improvement in accuracy of multiple sequence alignment. *Nucl Acids Res* 2005, 33(2):511-518.

9. Do CB, Mahabhashyam MSB, Brudno M, Batzoglou S: ProbCons: Probabilistic consistency-based multiple sequence alignment. *Genome Res* 2005, 15(2):330-340.

10. Roshan U, Livesay DR: Probalign: multiple sequence alignment using partition function posterior probabilities. *Bioinformatics* 2006, 22(22):2715-2721.

11. Clayton D, Hills M: Statistical models in epidemiology. Oxford ; New York: Oxford University Press; 1993.

12. Feng DF, Doolittle RF: Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J Mol Evol* 1987, 25(4):351-360.

13. Thompson JD, Higgins DG, Gibson TJ: CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research* 1994, 22(22):4673-4680.

14. Swofford DL, Olsen GJ: Phylogeny Reconstruction. In: *Molecular Systematics.* Edited by Hillis D, Moritz C, Marble BK, 2 edn. Sunderland, Massachusetts, USA: Sinauer Ass. Inc.; 1996: 407-514.

15. Morrison DA, Ellis JT: Effects of nucleotide sequence alignment on phylogeny estimation: a case study of 18S rDNAs of apicomplexa. *Mol Biol Evol* 1997, 14(4):428-441.

16. Goldman N: Effects of sequence alignment procedures on estimates of phylogeny. *BioEssays* 1998, 20:287-290.

17. Hall BG: Comparison of the accuracies of several phylogenetic methods using protein and DNA sequences. *Mol Biol Evol* 2005, 22(3):792-802.

18. Gotoh O: Significant improvement in accuracy of multiple protein sequence alignments by iterative refinement as assessed by reference to structural alignments. *J Mol Biol* 1996, 264(4):823-838.

19. Swofford DL: PAUP*: Phylogenetic analysis using parsimony (and other methods). In., 4.0 edn. Sunderland, Massachusetts: Sinaeur Associates; 1996.

20. Sjolander K: Phylogenomic inference of protein molecular function: advances and challenges. *Bioinformatics* 2004, 20(2):170-179.

21. Stoye J, Evers D, Meyer F: Rose: generating sequence families. *Bioinformatics* 1998, 14(2):157-163.

22. Hasegawa M, Kishino H, Yano T: Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. *J Mol Evol* 1985, 22(2):160-174.

23. Robinson DF, Foulds LR: Comparison of phylogenetic trees. *Mathematical Biosciences* 1981, 53:131-147.

24. La D, Sutch B, Livesay DR: Predicting protein functional sites with phylogenetic motifs. *Proteins* 2005, 58(2):309-320.

25. Sanderson MJ: r8s software package available from http://ginger.ucdavis.edu/r8s/, 2004.

26. Nakhleh L, Roshan U, Vawter L, Warnow T: Estimating the Deviation from a Molecular Clock. In: *Proceedings of the Second International Workshop on Algorithms in Bioinformatics.* Springer-Verlag; 2002: 287-299.

27. Wheeler W: POY: the optimization of alignment characters. Version 3.0.4. *Program and Documentation*, Available from ftp.amnh.org/pub/molecular, New York, NY; 2002.

28. Redelings BD, Suchard MA: Joint Bayesian estimation of alignment and phylogeny. *Syst Biol* 2005, 54(3):401-418.

29. Notredame C: Recent progress in multiple sequence alignment: a survey. *Pharmacogenomics* 2002, 3(1):131-144.

30. Thompson JD, Higgins DG, Gibson TJ: ClustalW: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research* 1994, 27:2682-2690.

31. Thompson JD, Koehl P, Ripp R, Poch O: BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark. *Proteins* 2005, 61(1):127-136.

32. Mizuguchi K, Deane CM, Blundell TL, Overington JP: HOMSTRAD: a database of protein structure alignments for homologous families. *Protein Sci* 1998, 7(11):2469-2471.

33. Raghava GP, Searle SM, Audley PC, Barber JD, Barton GJ: OXBench: a benchmark for evaluation of protein multiple sequence alignment accuracy. *BMC Bioinformatics* 2003, 4:47.

34. Edgar RC: MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucl Acids Res* 2004, 32(5):1792-1797.

35. Mattick JS, Makunin IV: Non-coding RNA. *Hum Mol Genet* 2006, 15 Spec No 1:R17-29.

36. Mehler MF, Mattick JS: Non-coding RNAs in the nervous system. *J Physiol* 2006, 575(Pt 2):333-341.

37. Eddy SR: Profile hidden Markov models. *Bioinformatics* 1998, 14(9):755-763.

38. Nawrocki EP, Eddy SR: Query-dependent banding (QDB) for faster RNA similarity searches. *PLoS Comput Biol* 2007, 3(3):e56 [http://infernal.janelia.org]. Infernal version 0.72.

39. Gardner PP, Wilm A, Washietl S: A benchmark of multiple sequence alignment programs upon structural RNAs. *Nucleic Acids Res* 2005, 33(8):2433-2439.

40. Smith TF, Waterman MS: Identification of common molecular subsequences. *J Mol Biol* 1981, 147(1):195-197.

41. Freyhult EK, Bollback JP, Gardner PP: Exploring genomic dark matter: a critical assessment of the performance of homology search methods on noncoding RNA. *Genome Res* 2007, 17(1):117-125.

42. Iyer S, Deutsch K, Yan X, Lin B: Batch RNAi selector: a standalone program to predict specific siRNA candidates in batches with enhanced sensitivity. *Comput Methods Programs Biomed* 2007, 85(3):203-209.

43. Klein RJ, Eddy SR: RSEARCH: finding homologs of single structured RNA sequences. *BMC Bioinformatics* 2003, 4:44.

44. Numata K, Kanai A, Saito R, Kondo S, Adachi J, Wilming LG, Hume DA, Hayashizaki Y, Tomita M: Identification of putative noncoding RNAs among the RIKEN mouse full-length cDNA collection. *Genome Res* 2003, 13(6B):1301-1306.

45. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ: Basic local alignment search tool. *J Mol Biol* 1990, 215(3):403-410.

46. Miyazawa S: A reliable sequence alignment method based on probabilities of residue correspondences. *Protein Eng* 1995, 8(10):999-1009.

47. Muckstein U, Hofacker IL, Stadler PF: Stochastic pairwise alignments. *Bioinformatics* 2002, 18 Suppl 2:S153-160.

48. Cochrane G, Aldebert P, Althorpe N, Andersson M, Baker W, Baldwin A, Bates K, Bhattacharyya S, Browne P, van den Broek A *et al*: EMBL Nucleotide Sequence Database: developments in 2005. *Nucleic Acids Res* 2006, 34(Database issue):D10-15.

49. Griffiths-Jones S, Moxon S, Marshall M, Khanna A, Eddy SR, Bateman A: Rfam: annotating non-coding RNAs in complete genomes. *Nucleic Acids Res* 2005, 33(Database issue):D121-124.

50. Personal communication with Alex Bateman of the RFAM database team.

51. Rivas E, Eddy SR: Noncoding RNA gene detection using comparative sequence analysis. *BMC Bioinformatics* 2001, 2:8.

52. Washietl S, Hofacker IL, Stadler PF: Fast and reliable prediction of noncoding RNAs. *Proc Natl Acad Sci USA* 2005, 102(7):2454-2459.

53. Coventry A, Kleitman DJ, Berger B: MSARI: multiple sequence alignments for statistical detection of RNA secondary structure. *Proc Natl Acad Sci USA* 2004, 101(33):12102-12107.

54. Phuong TM, Do CB, Edgar RC, Batzoglou S: Multiple alignment of protein sequences with repeats and rearrangements. *Nucleic Acids Res* 2006, 34(20):5932-5942.

55. Grice JA, Hughey R, Speck D: Reduced space sequence alignment. *Comput Appl Biosci* 1997, 13(1):45-53.

56. RNA-genome alignment benchmark and tools website, http://www.cs.njit.edu/usman/RNAgenome, Accessed: September 15, 2008.

57. Mount DW: Bioinformatics: sequence and genome analysis, 2 edn. Cold Spring Harbor, NY: Cold Spring Harbor Laboratory Press; 2006.

58. Pearson WR: Comparison of methods for searching protein sequence databases. *Protein Sci* 1995, 4(6):1145-1160.

59. Hulsen T, de Vlieg J, Leunissen JA, Groenen PM: Testing statistical significance scores of sequence comparison methods with structure similarity. *BMC Bioinformatics* 2006, 7:444.

60. Kanji GK: 100 statistical tests, 3rd edn. London ; Thousand Oaks, Calif.: Sage Publications; 2006.

61. Gribskov M, Robinson NL: Use of receiver operating characteristic (ROC) analysis to evaluate sequence matching. *Comput Chem* 1996, 20(1):25-33.

62. Alpaydin E: Introduction to machine learning. Cambridge, Mass.: MIT Press; 2004.

63. Consortium TD: Known type 1 diabetes associated regions, 2009.

64. Cornelis MC, Qi L, Zhang C, Kraft P, Manson J, Cai T, Hunter DJ, Hu FB: Joint Effects of Common Genetic Variants on the Risk for Type 2 Diabetes in U.S. Men and Women of European Ancestry. *Ann Intern Med* 2009, 150(8):541-550.

65. Durrant C, Zondervan KT, Cardon LR, Hunt S, Deloukas P, Morris AP: Linkage disequilibrium mapping via cladistic analysis of single-nucleotide polymorphism haplotypes. *American Journal of Human Genetics* 2004, **75**:35-43.

66. Evans DM, Visscher PM, Wray NR: Harnessing the information contained within genome-wide association studies to improve individual prediction of complex disease risk. *Human Molecular Genetics* 2009, 18(18):3525-3531.

67. Gail MH: Discriminatory accuracy from single-nucleotide polymorphisms in models to predict breast cancer risk. *N Engl J Med* 2008, 100(14):1037-1041.

68. Goldstein DB: Common genetic variation and human traits. *N Engl J Med* 2009, 360:1696-1698.

69. Guyon I, Weston J, Barnhill S, Vapnik V: Gene selection for cancer classification using support vector machines. *Machine Learning* 2002, 46:389-422.

70. Hardin D, Tsamardinos I, Aliferis CF: A theoretical characterization of linear SVM-based feature selection. In: *ICML '04: Proceedings of the twenty-first international conference on Machine learning.* New York, NY, USA: ACM; 2004: 48.

71. Hoggart CJ, Whittaker JC, De Iorio M, Balding DJ: Simultaneous Analysis of All SNPs in Genome-Wide and Re-Sequencing Association Studies. *PLoS Genet* 2008, 4(7):e1000130.

72. Hulbert EM, Smink LJ, Adlem EC, Allen JE, Burdick DB, Burren OS, Cavnor CC, Dolman GE, Flamez D, Friery KF *et al*: T1DBase: integration and presentation of complex data for type 1 diabetes research. *Nucl Acids Res* 2007, 35(1):D742-746.

73. Jakobsdottir J, Gorin MB, Conley YP, Ferell RE, Weeks DE: Interpretation of genetic association studies: markers with replicated highly significant odds ratios may be poor classifiers. *PLoS Genetics* 2009, 5.

74. Jewell NP: Statistics for Epidemiology: Chapman & Hall; 2003.

75. Joachims T: Making large-scale SVM learning practical. In: *Advances in Kernel Methods - Support Vector Learning.* MIT Press; 1999.

76. Kathiresan S, Melander O, Anevski D, Guiducci C, Burtt NP, Roos C, Hirschhorn JN, Berglund G, Hedblad B, Groop L *et al*: Polymorphisms Associated with Cholesterol and Risk of Cardiovascular Events. *N Engl J Med* 2008, 358(12):1240-1249.

77. Kraft P, Hunter DJ: Genetic Risk Prediction - Are We There Yet? *N Engl J Med* 2009, 360(17):1701-1703.

78. Li C, Li M: GWAsimulator: a rapid whole-genome simulation program. *Bioinformatics* 2008, 24(1):140-142.

79. Li C, Li M, ad R. M. Watanabe EML: Prioritized subset analysis: improving power in genome-wide association studies. *Hum Hered* 2008, 65:129-141.

80. Li J: Prioritize and select SNPs for association studies with multi-stage designs. *Journal of Computational Biology* 2007, 15(3):241-257.

81. Li Q, Yu K, Li Z, Zheng G: MAX-rank: a simple and robust genome-wide scan for case-control association studies. *Human Genetics* 2008, 123(6):617-623.

82. Macpherson M, Naughton B, Hsu A, Mountain J: 23andme.com white paper, 2007.

83. Mao W, Mao J: The application of random forest in genetic case-control studies. In: *Proceedings of International Conference on Technology and Applications in Biomedicine.* IEEE; 2008: 370-373.

84. Marchini J, Donnelly P, Cardon LR: Genome-wide strategies for detecting multiple loci that influence complex diseases. *Nature Genetics* 2005, 37(4):413-417.

85. Meng Y, Yu Y, Cupples LA, Farrer L, Lunetta K: Performance of random forest when SNPs are in linkage disequilibrium. *BMC Bioinformatics* 2009, 10(1):78.

86. Morrison AC, Bare LA, Chambless LE, Ellis SG, Malloy M, Kane JP, Pankow JS, Devlin JJ, Willerson JT, Boerwinkle E: Prediction of Coronary Heart Disease Risk using a Genetic Risk Score: The Atherosclerosis Risk in Communities Study. *Am J Epidemiol* 2007:kwm060.

87. Navigenics: Navigenics white paper, 2009.

88.

89.

90. Paynter NP, Chasman DI, Buring JE, Shiffman D, Cook NR, Ridker PM: Cardiovascular Disease Risk Prediction With and Without Knowledge of Genetic Variation at Chromosome 9p21.3. *Annals of Internal Medicine* 2009, 150.

91. Price AL, Patterson NJ, Plenge RM, Weinblatt ME, Shadick NA, Reich D: Principal components analysis corrects for stratification in genome-wide association studies. *Nature Genetics* 2006, 38:904-909.

92. Purcell S: PLINK v1.05 available at http://pngu.mgh.harvard.edu/purcell/plink/, 2009.

93. Purcell S, Neale B, Todd-Brown K, Thomas L, Ferreira MAR, Bender D, Maller J, Sklar P, de Bakker PIW, Daly MJ *et al*: PLINK: a toolset for whole-genome association and population-based linkage analysis. *American Journal of Human Genetics* 2007, 81.

94. Scholkopf B, Smola AJ: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. Cambridge, MA, USA: MIT Press; 2001.

95. Statnikov A, Hardin D, Aliferis C: Using SVM Weight-Based Methods to Identify Causally Relevant and Non-Causally Relevant Variables. In: *Proceedings of Neural Information Processing Systems (NIPS) Workshop on Causality and Feature Selection.* 2006.

96. Teo CH, Smola AJ, Vishwanathan SV, Le QV: A scalable modular convex solver for regularized risk minimization. In: KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining; San Jose, California, USA: ACM; 2007: 727-736.

97. Vapnik VN: The nature of statistical learning theory, 2nd edn. New York: Springer; 2000.

98. Wei Z, Sun W, Wang K, Hakonarson H: Multiple testing in genome-wide association studies via hidden Markov models. *Bioinformatics* 2009, 25(21):2802-2808.

99. Wellcome Trust Case Control Consortium: Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature* 2007, 447:661-678.

100. Wray NR, Goddard ME, Visscher PM: Prediction of individual genetic risk to disease from genome-wide association studies. *Genome Research* 2007, 17(10):1520-1528.

101. Zaykin D, Zhivotovsky L: Ranks of genuine associations in whole genome scans. *Genetics* 2005, 171(14):1037-1041.

102. Zheng G, Joo J, Lin JP, Stylianou M, Waclawiw MA, Geller NL: Robust ranks of true associations in genome-wide case-control association studies. *BMC Proc* 2007, 1 Suppl 1:S165.

103. Zheng W, Zou C, Zhao L: Weighted maximum margin discriminant analysis with kernels. *Neurocomputing* 2005, 67:357-362.

104. Zollner S, Pritchard JK: Overcoming the winner's curse: Estimating penetrance parameters from case control data. *Human Genetics* 2007, 80:605-615.

105. Durbin R: Biological sequence analysis : probabalistic models of proteins and nucleic acids. Cambridge, UK New York: Cambridge University Press; 1998.

106. Dayhoff MO, Schwartz RM, Orbutt BC: A model of evolutionary changes in proteins. In: *Atlas of protein sequence and structure* Edited by Dayhoff MO, vol. 5. Washington, DC: National Biomedical Research Foundation; 1978: 345-352.

107. Altschul SF: A protein alignment scoring system sensitive at all evolutionary distances. *J Mol Evol* 1993, 36(3):290-300.

108. Karlin S, Altschul SF: Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proc Natl Acad Sci U S A* 1990, 87(6):2264-2268.