Flexible Discovery of Modules with Distance Constraints

Øystein Lekang

Master of Science in Computer Science
Submission date: June 2006
Supervisor: Arne Halaas, IDI
Co-supervisor: Geir Kjetil Sandve, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science
Problem Description

Many authors argue that finding single transcription factor binding sites is not enough to be able to make predictions with regard to regulation in eukaryotic genes, as is the case with simpler prokaryotes. With eukaryotes combinations of transcription factors must be modeled as a composite motif or module. In some cases even with a restriction on distance between individual sites, or within the module. Create a module discovery tool capable of using both deterministic patterns and position weight matrices as input, that can impose restrictions on distance. Use the tool for module discovery and evaluate the results.

Supervisor: Arne Halaas, IDI
Abstract

Currently, whole genomes have been sequenced and the next big challenge is to map the functional elements. The essence of this challenge is finding specifics in a vast amount of information, and it is therefore natural for the fields of biology and information technology to cooperate in creating solutions to meet it. The field of motif discovery has in recent years seen a shift of attention from single motifs to sets of motifs that cooperate in regulating the expression of a gene. This has resulted in a new and more challenging objective. Instead of finding single transcription factor binding sites (TFBS) we now have to tackle the problem of finding collections of TFBSs modelled as composite motifs (modules). The objective of this project is to create a tool that can use Position Weight Matrices as input, is able to impose distance restrictions on the composite motifs generated, and lends itself to further development.

To validate the functionality of the tool two large datasets were used. The first was from the TransCompel database and contains combinations of motifs known to span a short distance. With this dataset evaluation was performed on a per nucleotide position basis. The other was ChIP-CHIP-data from yeast (Saccharomyces), where the presence of modules was evaluated.

With TransCompel data the tool achieved a relatively low but positive correlation. It was also able to discern a higher correlation with distance restricted. The low correlation is mainly caused by problems related to finding matches for the single motifs in the sequences.

On yeast data the expectations were not high, given that the dataset is based on ChIP-CHIP experiments. This also causes the dataset to lack positional information, and the evaluation was performed on presence alone. The tool achieved a positive correlation on these data, and found an indication that distances may be involved.
Contents

1 Introduction 1

2 Biology 3
  2.1 Biological background and concepts 3
    2.1.1 Transcription 3
    2.1.2 Regulation 4
    2.1.3 Transcription factor binding sites 4
  2.2 Identifying TFBS 5

3 Motif Discovery 7
  3.1 Motif representation 7
    3.1.1 Deterministic Patterns 7
    3.1.2 Position Weight Matrix 8
    3.1.3 Other Models 9
  3.2 Discovering TFBS 10
    3.2.1 Prior Knowledge and Limiting Searchspaces 10
  3.3 Modules 10
    3.3.1 Distance Restrictions in Modules 11
  3.4 GCMD 12
  3.5 MOP 12
  3.6 Evaluation Statistics 14

4 Design and Implementation 17
  4.1 Motifs and Variants 17
  4.2 Distance 18
  4.3 Significance 20
    4.3.1 Regular Expressions 20
    4.3.2 PWM 20
  4.4 The Pareto Front 22
  4.5 Composite Search 24
  4.6 Pruning 24
  4.7 Evaluation 25
  4.8 Choice of Language 25
    4.8.1 Conclusion 27
    4.8.2 Language Limitations 27
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 Usage of the diplom tool</td>
<td>29</td>
</tr>
<tr>
<td>5.1 Input</td>
<td>29</td>
</tr>
<tr>
<td>5.2 Output</td>
<td>30</td>
</tr>
<tr>
<td>5.3 Configuration</td>
<td>30</td>
</tr>
<tr>
<td>6 Results</td>
<td>33</td>
</tr>
<tr>
<td>6.1 TransCompel Datasets</td>
<td>33</td>
</tr>
<tr>
<td>6.2 Yeast Datasets</td>
<td>33</td>
</tr>
<tr>
<td>6.3 TransCompel Results</td>
<td>34</td>
</tr>
<tr>
<td>6.3.1 100% sets</td>
<td>35</td>
</tr>
<tr>
<td>6.3.2 50% sets</td>
<td>37</td>
</tr>
<tr>
<td>6.3.3 25% set</td>
<td>39</td>
</tr>
<tr>
<td>6.4 Yeast Results</td>
<td>41</td>
</tr>
<tr>
<td>7 Discussion</td>
<td>43</td>
</tr>
<tr>
<td>7.1 TransCompel</td>
<td>43</td>
</tr>
<tr>
<td>7.2 Yeast</td>
<td>43</td>
</tr>
<tr>
<td>7.3 Algorithm</td>
<td>44</td>
</tr>
<tr>
<td>Bibliography</td>
<td>44</td>
</tr>
<tr>
<td>A Efficiency Test on Bitsets</td>
<td>49</td>
</tr>
<tr>
<td>A.1 Results and Description</td>
<td>49</td>
</tr>
<tr>
<td>A.2 Source Code</td>
<td>49</td>
</tr>
<tr>
<td>B Manual page</td>
<td>55</td>
</tr>
<tr>
<td>C Complete TransCompel Results</td>
<td>57</td>
</tr>
</tbody>
</table>
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Double helix of DNA, displaying all 4 nucleotides</td>
<td>3</td>
</tr>
<tr>
<td>2.2</td>
<td>Coding area for a gene, including the promoter region</td>
<td>4</td>
</tr>
<tr>
<td>3.1</td>
<td>Most common motif representations</td>
<td>8</td>
</tr>
<tr>
<td>3.2</td>
<td>HHM-models</td>
<td>9</td>
</tr>
<tr>
<td>3.3</td>
<td>Coregulated concept</td>
<td>11</td>
</tr>
<tr>
<td>3.4</td>
<td>Ideal solution for MOP</td>
<td>13</td>
</tr>
<tr>
<td>3.5</td>
<td>Pareto Front for bi-objective problem</td>
<td>14</td>
</tr>
<tr>
<td>4.1</td>
<td>The motifs</td>
<td>17</td>
</tr>
<tr>
<td>4.2</td>
<td>Motif with variants</td>
<td>18</td>
</tr>
<tr>
<td>4.3</td>
<td>Distance Class</td>
<td>19</td>
</tr>
<tr>
<td>4.4</td>
<td>Visualisation of threshold calculation</td>
<td>21</td>
</tr>
<tr>
<td>4.5</td>
<td>Visualisation of the effect of Pareto optimality</td>
<td>22</td>
</tr>
<tr>
<td>4.6</td>
<td>Multiple number of components</td>
<td>23</td>
</tr>
<tr>
<td>4.7</td>
<td>Pareto with components and distances</td>
<td>23</td>
</tr>
<tr>
<td>4.8</td>
<td>Visualisation of Search</td>
<td>26</td>
</tr>
<tr>
<td>6.1</td>
<td>Achieved Coverage on a Positive Set</td>
<td>34</td>
</tr>
<tr>
<td>6.2</td>
<td>Position Evaluation</td>
<td>35</td>
</tr>
<tr>
<td>6.3</td>
<td>Correlation Coefficients, TP, 100%</td>
<td>35</td>
</tr>
<tr>
<td>6.4</td>
<td>Correlation Coefficients, FP, 100%</td>
<td>36</td>
</tr>
<tr>
<td>6.5</td>
<td>Correlation Coefficients, FP, 100%</td>
<td>36</td>
</tr>
<tr>
<td>6.6</td>
<td>Correlation Coefficients, TP, 50%</td>
<td>37</td>
</tr>
<tr>
<td>6.7</td>
<td>Correlation Coefficients, FP, 50%</td>
<td>38</td>
</tr>
<tr>
<td>6.8</td>
<td>Correlation Coefficients, FP, 50%</td>
<td>38</td>
</tr>
<tr>
<td>6.9</td>
<td>Correlation Coefficients, TP, 25%</td>
<td>39</td>
</tr>
<tr>
<td>6.10</td>
<td>Correlation Coefficients, FP, 25%</td>
<td>40</td>
</tr>
<tr>
<td>6.11</td>
<td>Correlation Coefficients, FP, 25%</td>
<td>40</td>
</tr>
</tbody>
</table>
List of Tables

3.1 Tools and algorithms for motif discovery . . . . . . . . . . . . . . . . 10
4.1 Language efficiency . . . . . . . . . . . . . . . . . . . . . . . . . . 27
6.1 Results from likelihoods set to 0.8, 0.9 and 1.0 . . . . . . . . . . . . . 35
6.2 Results on 100% set . . . . . . . . . . . . . . . . . . . . . . . . . . 36
6.3 Results from likelihoods set to 0.8, 0.9 and 1.0 . . . . . . . . . . . . . 37
6.4 Results on 50% set . . . . . . . . . . . . . . . . . . . . . . . . . . 38
6.5 Results from likelihoods set to 0.8, 0.9 and 1.0 . . . . . . . . . . . . . 39
6.6 Results on 25% set . . . . . . . . . . . . . . . . . . . . . . . . . . 40
6.7 Yeast results 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . 41
6.8 Yeast results 2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . 41
6.9 Yeast results 3 . . . . . . . . . . . . . . . . . . . . . . . . . . . . 42
A.1 Efficiency test of bitsets . . . . . . . . . . . . . . . . . . . . . . . . 49
Chapter 1

Introduction

In recent years the research into discovery of transcription factor binding sites (TFBS) has received much attention. Lately the focus has been mainly to the idea of modules. That is, a collection of TFBS in the upstream region of a gene that together affect regulation of that gene. A multitude of applications have been created for the purpose of finding TFBS. Modules have been modelled by a wide variety of models, ranging from complex combinatorial and hidden markov models to simple set models. Many such models are equipped with some kind of restraint on distance. This is done under the assumption that physical nearness may be a factor if the motifs affect regulation together. Distance restraints are most common where the composite motif is modelled as a tuple of single motifs. In that case distance restraints can be imposed on the distance between motifs or spanning the whole module.

In this project the distance restriction is a maximum window. The module is modelled as a set, so as not to introduce complexity that could impose restrictions other than that on distance. This way results are directly connected to the distance restraint with as few other factors as possible. To make the tool flexible with regard to what data it can use, it is equipped with capability to use both deterministic patterns (regular expressions) and probabilistic position weight matrices.
Chapter 2

Biology

This chapter describes the biological background. It starts with a short introduction to some of the biological concepts concerning regulation. After this it describes some of the wetlab tools and methods used to find functional elements in the Deoxyribonucleic acid (DNA). These are presented because they are often the methods used as input, premise or control in for computerised methods.

2.1 Biological background and concepts

For computer scientists DNA is a string over the alphabet $\sum$. Figure 2.1 displays the double helix of DNA. Each strand with its nucleotides corresponding to the nucleotide on the other strand. It is included here to visualise the complexity of the actual systems, compared to the models used to understand and analyse them.

Figure 2.1: Double helix of DNA, displaying all 4 nucleotides

2.1.1 Transcription

Transcription is the process whereby the DNA sequence in a gene is copied into messenger-RNA (mRNA). This process of copying the gene can be said to be performed by an
2.1. BIOLOGICAL BACKGROUND AND CONCEPTS

CHAPTER 2. BIOLOGY

An enzyme called RNA-polymerase. RNA-polymerase is attracted by signals bound to binding sites denoted by “promotor” and “enhanser” in figure 2.2. These signals are called transcription factors (see section 2.1.3). The transcription starts at a “transcription start site” (TSS) at the beginning of the coding region. There may be more than one TSS for each gene. Thereafter the nucleotides in the coding region are coupled with the corresponding creating the mRNA. The mRNA may then go through “splicing” where the introns are removed, and may be transported out of the nucleus to be translated into a protein. These processes are however outside the scope of this project, as it’s focus is on transcription and the gene regulation occurring through transcription factors.

![Figure 2.2: Coding area for a gene, including the promoter region.](image)

2.1.2 Regulation

An important concept in this report is gene regulation. “Regulation of gene expression” or “gene regulation.” Regulation is the amount or timing of introduction of a functional product of a gene. The product can be anything from mRNA to a protein or even a modified protein. In this text the term is used as a descriptor of the affinity to transcribe a given gene. An up regulated gene has higher likelihood of being transcribed and is transcribed more often.

2.1.3 Transcription factor binding sites

Transcription Factor is a description of proteins that attach themselves to binding sites in a gene’s promoter region, and thereby regulate the transcription of the gene. For this reason the discovery of such functional elements in the non-coding regions of DNA can be very valuable. The regulation of a gene is much more complex than this description implies. Orientation and folding of the DNA are significant, among many other factors. Still the transcription factor binding sites are two of the most important functional elements in any genome, for understanding regulation. TFBSs are usually short, around 5-15 base pairs (bp), and are frequently degenerate sequence motifs (figure 3.1). As a result of this potential binding sites can occur frequently by chance in large genomes of higher eukaryotes. This makes the functional elements even harder to find.


2.2 Identifying TFBS

How the genome encodes information that specifies when and where a gene will be expressed is a major challenge in interpreting genome sequences. This process has to be started by identifying regions of the genome that contain regulatory information. Traditionally discovering binding information for TFs has been done through time-consuming approaches. These were either analysis that monitors the change in mobility when DNA and protein bind, footprinting that identifies the region of DNA protected by protein, or reporter constructs. In recent years, however, much effort has been directed to develop technologies that have a high throughput. Methods have been developed for this purpose for in vitro (SELEX) and in vivo (Microarray, ChIP-ChIP) identification.

These high throughput methods have made genome sequences available for a number of species and thereby opened the door for computational methods.
Chapter 3

Motif Discovery

Computerised methods for analysing biological data have been developing for years. Now that whole genomes have been sequenced they are more promising than ever. This chapter will start by describing how motifs usually are represented when treated \textit{in silico}. Priority is lent to the topics that are used in this project. Then some general topics concerning motif discovery will be covered, along with a description of the tool “General Composite Motif Discovery” from which this project stems. Lastly some evaluation values are listed for reference.

3.1 Motif representation

This section gives a short introduction to the two most common representations of motifs. They are used in publicly available databases like Prosite\cite{prosite}, Transfac\cite{transfac}, and Jaspar\cite{jaspar}.

3.1.1 Deterministic Patterns

The simplest kind of deterministic pattern is just a sequence of characters from the alphabet $\sum$. In terms of DNA this would be the alphabet \{A, C, G, T\}. On top of this there are multiple expansions like \textit{IUPAC extensions} and \textit{regular expressions} or more generally ambiguities, wildcards and gaps. Consensus strings are strings over the alphabet $\sum$ that captures the composition of most occurrences of the motif. This consensus string is often expanded either by regular expressions or IUPAC strings\cite{iupac}. Regular expression subsets most used are braces for ambiguities or a dot for “any character” while IUPAC has different symbols for the different combinations of bases/nucleotides. Figures \ref{fig:regexp} to \ref{fig:iupac} displays the different variants. The reg.exp. and IUPAC string are not completely equal since the IUPAC “N” equals the “.” or [ACGT] in regular expressions.

\textbf{Significance} With string representations of motifs are usually logods score over a markov background. This assumes independence between nucleotides in varying degree depending on the order of the markov chain. The reason for using logarithms is purely computational. With logarithms the calculation is mainly addition instead of multiplication with probabilities, thereby fewer instructions. It also removes trouble with accuracy on small likelihoods.
3.1. Motif Representation

3.1.2: Position Weight Matrix

A Position Weight Matrix is a matrix of score values that gives a weighted match to any given substring of fixed length. In DNA each row will represent a nucleotide, and the columns are positions in the substring. The PWM is created from an alignment matrix (count matrix) where the count of each nucleotide in each position when aligned is recorded. From the alignment matrix a probability matrix can be created (see equation 3.1). This is a matrix of the likelihood weighted on background residue. In this probability matrix all columns or “positions” \( \sum_{i=1}^{A} f_{i,j} = 1 \) since each position must have a character from the given alphabet \( A \). Equation 3.2 results in the PWM. An example is shown in figure 3.1.4 it is created from the sequences in figure 3.1.1.

\[
f_{i,j} = \frac{n_{i,j} + p_i k}{\sum_{i=1}^{A} n_{i,j} + k}
\]  
(3.1)

\[
W_{i,j} = \ln \left( \frac{f_{i,j}}{p_i} \right)
\]  
(3.2)

- \( f_{i,j} \): corrected frequency of residue \( i \) at position \( j \)
- \( A \): alphabet size
- \( p_i \): prior/background probability for residue \( i \)
- \( k \): pseudo weight (arbitrary)

The PWM can then be used to scan sequences for hits by using a sliding window, and calculating the score for each position of the window. The score is equivalent to the logarithm and the ratio between the product of the matrix frequencies and the product of the prior probabilities of the residues found in the sequence segment. This can also
be thought of as a logarithm of the ratio of the probabilities of being within or without the motif.

**Significance** for PWMs are calculated in multiple ways. One common solution is to calculate the **Information Content** of the matrix (see equation 3.3 $w$ is the length of the matrix).

\[
f_{i,j} = \frac{n_{i,j} + p_i k}{\sum_{i=1}^{A} n_{i,j} + k}
\]

\[
I = \sum_{j=1}^{w} \sum_{i=1}^{A} f_{i,j} \ln \frac{f_{i,j}}{p_i}
\] (3.3)

Another way is to use the background likelihood, or p-value. This is most often done by matching the matrix on either a real background [37] or one generated from a markov model [4, 37]. It can also be calculated directly from the background model [41]. See section 3.6 for a more detailed description of significance.

### 3.1.3 Other Models

There are multiple other models, specially in machine learning approaches. An example Hidden Marker Models (HHM). A HMM is a statistical model that assumes the system modelled is a Markov process with unknown parameters. In a Markov process the probability distribution of future states, depends solely on the current state. The challenge is to determine the hidden parameters, from the observable parameters. The extracted model parameters can then be used to perform further analysis, for example for pattern recognition applications. Figure 3.2.2 is a commonly used HHM topology for sequence analysis [3, 35]. It allows for insertions and deletions, or gaps and mismatches. Figure 3.2.1 displays the topology of meta-MEME, where only insertions are allowed and only between single motifs.

![Figure 3.2](image)

**Figure 3.2:** Composite motif model from meta-MEME, and a general HHM for profiles


### 3.2 Discovering Transcription Factor Binding Sites

Pattern discovery and classification have been in use for a long time, even in a biological context. This has resulted in a wide variety of methods for computer sciences being applied to the field. Table 3.1 lists some approaches.

<table>
<thead>
<tr>
<th>Machine Learning</th>
<th>Heuristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neural Networks</td>
<td>Gibbs sampling</td>
</tr>
<tr>
<td>Blekas et al. [9] [8]</td>
<td>AlignAce [19]</td>
</tr>
<tr>
<td>Gulukota et al. [17]</td>
<td>BioProspector [25]</td>
</tr>
<tr>
<td>Support Vector Machine</td>
<td>Expectation Max.</td>
</tr>
<tr>
<td>Eveillard et al. [14]</td>
<td>MEME [7]</td>
</tr>
<tr>
<td>Hidden Markov Model</td>
<td>Greedy Algorithm</td>
</tr>
<tr>
<td>Krogh et al. [3] [20]</td>
<td>CONSENSUS [18]</td>
</tr>
<tr>
<td>Meta-MEME [16]</td>
<td>Exact</td>
</tr>
<tr>
<td></td>
<td>Enumeration</td>
</tr>
<tr>
<td>MOTIF [40]</td>
<td>Pratt [22]</td>
</tr>
<tr>
<td>Pratt [22]</td>
<td>TEIRESISAS [38]</td>
</tr>
</tbody>
</table>

**Table 3.1:** Tools and algorithms for motif discovery

### 3.2.1 Prior Knowledge and Limiting Searchspaces

The way one approaches motif discovery is largely dependent on what, and how much *a priori* knowledge that is available. One approach is to use the whole genome as the search space, and search for patterns in upstream regions that are statistically overrepresented. Alternatively one can combine computerized discovery with biological methods to limit the search space to “co regulated” genes or genes that we suspect may be regulated in a similar way. This way the search is confined to what we suspect to be a positive set. Another way to benefit from prior knowledge is that of RepeatMasker[5]. In the mid 1960’s scientists discovered that many genomes contain stretches of highly repetitive DNA sequences. These are largely believed to be nonfunctional. RepeatMasker is a program that removes repeats from the genome sequence, and thereby limiting the searchspace. Evolution can also help with the process. *Phylogenetic footprinting* is a method to identify regions of sequence conservation between genomes. The strategy is based on the assumption that evolution causes the important regulatory sites to be contained in these conserved regions. The technique has been described as searching for “islands of conserved sequences in seas of less conserved noncoding sequence”[24].

### 3.3 Modules or Composite Motifs

Computational discovery of single motifs has proven profitable, particularly when applied to sequences from prokaryotic organisms and yeast. Some cases however provide a rate of false positives that remains to high. Non-coding regions are much longer in
humans or mice compared to yeast. Kellis et. al. [27] states that yeast genomes are about 30% coding whereas the human genome is about 2% coding, and that intergenic regions of yeast consist of about 15% regulatory elements, while in human about 3% of intergenic regions are regulatory elements. This indicates the difficulty of extrapolating classic motif discovery techniques to higher eukaryotes. In addition it is believed that transcriptional regulation require the combinatorial interplay of multiple factors [10, 37, 23, 29]. See figure 3.3 for a simple conceptual picture.

Figure 3.3: Conceptual picture of multiple Transcription Factors regulating the transcription of a gene. In this case RNA polymerase binds to the factors TF_1 and TF_2, and they must therefore both be there to have an effect on regulation.

Sinha et. al. divides composite models into four groups [39]. These are the set model which is built on the assumption that the occurrence of a set of simple motifs is all that is needed for regulation. This is the least complex of the model categories but is believed to be close to a biological reality, with higher eukaryotes. The tuple model consists of ordered tuples of single motifs, with distance constraints on the gaps. The constraints on distance are usually tight intervals or fixed lengths. Hidden Markov Models (HHM) is the third of the group. HHMs consist of a number of states, with which an emission- and a transition probability distribution is associated. This enables walking through the states in the HHM emitting a nucleotide in each emission-state. (see figure 3.2.1) The last group is the Boolean that model composite motifs by boolean combinations of counts, distances or locations of simple motifs. i.e. given the simple motifs $m_1$ and $m_2$ a composite motif might be “$(\text{count}(m_1) > 2) \lor (\text{count}(m_2) \geq 1) \land \text{count}(m_1) = 1 \land \text{dist}(m_1, m_2) < d)$.” Which would specify the motif as wither more than 2 of $m_1$ or one $m_1$ and at least one $m_2$ with a limited distance between $m_1$ and $m_2$.

Approaches

With composite motif discovery one can either discover de-novo composite motifs directly, or use single motifs as input and combine them to composites. Some tools do both the novel single motif discovery and then use combinations of these for composite building.

The BioProspector program by Liu et. al. [25] is an example of finding the composite directly. It finds a tuple of two motifs with a ranged gap between them. It takes as input the width of the two blocks for the motifs, and a gap range. The algorithm then uses a modified Gibbs sampling strategy to find the composite.

3.3.1 Distance Restrictions in Modules

Many kinds of distance restrictions are in use. Of these strict or flexible gaps [25, 21, 13] and distance windows. Alternatively one could provide some function on the distance between motifs that premiers motifs separated with some distance and penalizes evenly as the distance grows or shrinks.
3.4 General Composite Motif Discovery

GCMD [15] is a tool that combines reg.exp motifs into composites. It belongs to the set model as occurrence is modeled by a bit for each sequence. This enables the tool to use these bitsets that represent occurrences in sequences to combine motifs through a logical “and” operation. The algorithm uses negative log likelihood for significance, thereby making maximization the optimal for both significance and support. It keeps track of the best composites in a pareto front, and uses the current front as a basis for pruning. Although this algorithm must be said to be a member of Sinhas set model it does however have the option to enforce order on the motifs in a composite. The worst case time complexity of GCMD is $O(n^c)$, where $n$ is the number of input motifs and $c$ is the desired number of components. In practice the pruning arrangement makes the algorithm run faster.

GCMD is the model on which this project is built. The algorithm is to be reimplemented with support for position weight matrices, and the ability to impose distance restraints on the composites it produces. The chosen distance model is a window defining the maximum length of the composite. The choice is made to keep the flexibility of a set model, and since the algorithm should be able to run multiple distances simultaneously it is not so much a restriction as an expansion of output information.

3.5 Multiobjective Optimization Problem

When performing motif discovery two important values are the coverage or support and the significance of motifs. These values are in many respects opposites. To increase one there is usually a fall in the other. This kind of optimization problem is referred to as a Multiobjective optimization problem. The Multiobjective Optimization Problem (MOP) was defined by Osyczka [31] as the problem of finding

\[
\text{a vector of decision variables which satisfies constraints and optimizes a vector function whose elements represent the objective functions. These functions form a mathematical description of performance criteria which are usually in conflict with each other. Hence, the term “optimize” means finding such a solution which would give the values of all the objective functions acceptable to the decision maker.}
\]

More formally this can be stated as follows [12]: The vector $\bar{x}^* = [x_1^*, x_2^*, \ldots, x_n^*]^T$ denotes the optimal solutions. (Usually more than one) Find the $\bar{x}^*$ that will satisfy the $m$ inequality constraints

\[
g_i(\bar{x}) \geq 0 \quad i = 1, 2, 3, \ldots, m \tag{3.4}
\]

the $p$ equality constraints

\[
h_i(\bar{x}) = 0 \quad i = 1, 2, 3, \ldots, m \tag{3.5}
\]

and optimizes the vector function

\[
\bar{f}(\bar{x}) = [f_1(\bar{x}), f_2(\bar{x}), \ldots, f_k(\bar{x})]^T \tag{3.6}
\]

where $\bar{x} = [x_1, x_2, x_3, \ldots, x_n]^T$ is the vector of decision variables.
The constraints of given equations (3.4) and (3.5) represent the restrictions imposed on the decision variables and defines the feasible region $F$ and any point $\hat{x} \in F$ defines a feasible solution.

If there is an $\hat{x}^*$ where

$$\forall x \in F \{ f_i(\hat{x}^*) \geq f_i(\hat{x}) \}$$

(3.7)

then the problem is trivial and has one solution. This is however rarely the case. Normally there is no one solution where all the $f_i(\hat{x})$ has their maximum in $F$ (see figure 3.4) at a common point $x^*$. 

![Figure 3.4: Ideal solution where all objectives has their maximum in one point, in a problem with two objective functions](image)

**Pareto Optimum** is a concept formulated by Vilfredo Pareto[34], and is considered the birth of MOP. A vector $\hat{x}^*$ can be considered Pareto Optimal if there exists no other vector $\hat{x} \in F$ that increases some criteria without decreasing another criteria. More formally stated, a vector of decision variables $\hat{x}^* \in F$ is Pareto Optimal if there exists no $\hat{x} \in F$ where

$$\forall i \in I \{ f_i(\hat{x}) \geq f_i(\hat{x}^*) \} \quad i = 1, 2, \ldots, k$$

(3.8)

or

$$\exists j \{ f_j(\hat{x}) > f_j(\hat{x}^*) \}$$

(3.9)

As already stated this, unfortunately, almost always gives more than one solution, but a set for nondominated solutions.

**Pareto Front** is the maxima in the Pareto sense. It is located in the boundary of the design region, or in the locus of the tangent points of the objective functions. In general this line or surface that contains these points is not easy to find. Figure 3.5 displays a bi-objective problem with the Pareto Front marked.

---

1In the literature authors usually prefer to use minimum as optimal, so the $\geq$ would be a $\leq$. 

13
3.6 Evaluation Statistics

The process of pattern matching can be viewed as a hypothesis testing, where the null hypothesis \((H_0)\) is that the observations are the result of pure chance, and the alternative hypothesis \((H_1)\) is that the observations show a real effect. Then the \(p - value\) can be computed, which is the probability that a test statistic at least as significant as the one observed would be obtained assuming that the null hypothesis were true. The smaller the \(p - value\) the stronger evidence against the null hypothesis. In a hypothesis test this \(p - value\) would be compared to an acceptable significance value \(\alpha\). If the \(p - value\) is at least as small as \(\alpha\) the null hypothesis is ruled out and the alternative hypothesis is valid.

If put in a pattern matching context \(H_0\) is that the sequence segment is noise (generated by the background), and \(H_1\) that it is a TFBS. The \(p - value\) would be the probability that the sequence segment is generated by the background model. This \(p - value\) is compared to a threshold which determines whether is a match is found.

Significance
The significance of motifs in this project is a negative logarithm (base 2) of the required \(p\)-value to constitute a hit. In other words a negative log likelihood over the background.

Symbol Explanation
These symbols are used for the calculation of the remaining statistical values.
- TP - True Positive
- FP - False Positive
- TN - True Negative
- FN - False Negative

Sensitivity
The probability that a statistical test will be positive for a true statistic, or that a
CHAPTER 3. MOTIF DISCOVERY 3.6. EVALUATION STATISTICS

TFBS is predicted where one in present.

\[ Sn = \frac{TP}{TP + FN} \]  \hspace{1cm} (3.10)

**Specificity**

The probability that a statistical test will be negative for a negative statistic, or that no TFBS is predicted where none is present.

\[ SP = \frac{TN}{TN + FP} \]  \hspace{1cm} (3.11)

**Possitive Predictive Value**

The probability that a test gives a true result for a true statistic.

\[ PPV = \frac{TP}{TP + FP} \]  \hspace{1cm} (3.12)

**Correlation Coefficient**

Also known as the Pearson product-moment coefficient of correlation in the particular case of two binary variables.

\[ CC = \frac{TP * TN - FN * FP}{\sqrt{(TP + FN)(TN + FP)(TP + FP)(TN + FN)}} \]  \hspace{1cm} (3.13)

**Performance Coefficient**

Proposed used in this context by Pevzner et.al. [33], and used in the model this project uses for evaluation [28].

\[ PC = \frac{TP}{TP + FN + FP} \]  \hspace{1cm} (3.14)

**Average Site Performance**

Proposed by Burset et.al. [26], also used in [28].

\[ ASP = \frac{Sn + PPV}{2} \]  \hspace{1cm} (3.15)
Chapter 4

Design and Implementation

This chapter describes the main challenges and chosen solutions for the implementation of the tool. It presents the datastructures and algorithms in various detail depending on their significance for the tool as a whole.

4.1 Motifs and Variants

The model of the motif in is largely inherited from GCMD\cite{15, 43} in that the search is based on bitwise operations on a bitset. It is also similar in that a significance is needed to drive the search. The support or coverage value is essentially the cardinality of the bitset, and is therefore available. In this project however PWMs are to be used as input in addition to regular expressions, and distance restriction are imposed. The inclusion of PWMs poses some interesting questions on how to compute the significance and support values, discussed in section 4.3. The inclusion of PWMs does however result in three different kinds of motifs. These being the two input types and the composite. They are essentially different in what they contain, although they share the values needed for the search. The motifs in this project are modeled as in figure 4.1.

![Figure 4.1: The different kinds of motifs and their relationship](Image)

Another model considered was to have the composite motif, just be an index of other motif models. This would be very easy to understand for reuse and generally elegant. It would however have to be implemented with caching, to avoid the calculation of resulting bitsets, significance and matches to sequences every time the motif is to be expanded or tested for values. These issues makes that model approximate the
one used. In addition there are some considerations concerning mutability with bitsets. The and operations work on the bitset, so a new bitset has to be created to represent the composite, else the original motif is changed. Also the python copy and deepcopy operations are unable to traverse into the java code. This could be remedied by implementing them, but it was decided that these considerations combined justifies the model with the composite as a subclass of motif.

A deciding factor on the general model of the motif is that it needs to incorporate distance restraints. This means keeping track of distinct hits, and a separate bitset and thereby support for each distance class. A distance class in this context refers to one of possibly several distance restraints that are run concurrently. Figure 4.1 shows the model. The choice of this solution over creating separate motifs for each distance lies in the amount of work to evaluate the motif distances, and is discussed int section 4.2.

4.2 Distance

The distance restrictions should not limit what the tool is able to find, and thereby foregoing the advantage in the flexibility of the set model. Instead it hoped to achieve a distance “categorisation” so that one can output optimal composites for each distance class, for comparison.

Distance restrictions are kept track of by a dedicated class, see figure 4.2 The primary function here is the to evaluate the required size of the distance window in a sequence for the motif. This is done by the function calc_dist_class. The function uses the hitlist of the motif and generates the permutations of hits in each sequence. The permutations are then evaluated and the tightest window and its position is kept, see algorithm 4.2.1 By arranging the different distance classes as variants of motifs, this calculation only has to be executed once for each motif, not for each variant. Variants receive the minimal distance for each sequence and can thereby determine their bitsets, and whether they are to continue their existence. The calculation also removes the possibility of overlaps between members of a composite.

Figure 4.2: The Motif model with variants. The figure only display variables and methods that are discussed.

4.2 Distance

The distance restrictions should not limit what the tool is able to find, and thereby foregoing the advantage in the flexibility of the set model. Instead it hoped to achieve a distance “categorisation” so that one can output optimal composites for each distance class, for comparison.

Distance restrictions are kept track of by a dedicated class, see figure 4.2. The primary function here is the to evaluate the required size of the distance window in a sequence for the motif. This is done by the function calc_dist_class. The function uses the hitlist of the motif and generates the permutations of hits in each sequence. The permutations are then evaluated and the tightest window and its position is kept, see algorithm 4.2.1. By arranging the different distance classes as variants of motifs, this calculation only has to be executed once for each motif, not for each variant. Variants receive the minimal distance for each sequence and can thereby determine their bitsets, and whether they are to continue their existence. The calculation also removes the possibility of overlaps between members of a composite.

Figure 4.2: The Motif model with variants. The figure only display variables and methods that are discussed.

4.2 Distance

The distance restrictions should not limit what the tool is able to find, and thereby foregoing the advantage in the flexibility of the set model. Instead it hoped to achieve a distance “categorisation” so that one can output optimal composites for each distance class, for comparison.

Distance restrictions are kept track of by a dedicated class, see figure 4.2. The primary function here is the to evaluate the required size of the distance window in a sequence for the motif. This is done by the function calc_dist_class. The function uses the hitlist of the motif and generates the permutations of hits in each sequence. The permutations are then evaluated and the tightest window and its position is kept, see algorithm 4.2.1. By arranging the different distance classes as variants of motifs, this calculation only has to be executed once for each motif, not for each variant. Variants receive the minimal distance for each sequence and can thereby determine their bitsets, and whether they are to continue their existence. The calculation also removes the possibility of overlaps between members of a composite.

1 A list containing matches for each sequence
**Algorithm 4.2.1: DISTANCE EVALUATION(permutations)**

```python
procedure CALC LENGTH(permutations)
local stub ← (0, 1)
local distance = MAX_INT
local window
for each perm ∈ permutations
local temp
for each position ∈ perm
do (comment: Add the start and end positions for the hit,
   comment: along with markers to a list
   temp.append(position, start, 0)
   temp.append(position, end, 1)
   temp.sort())
do (comment: Assert that there are no overlaps, using the markers
if temp.column(2) = stub * len(perm)
then
   (comment: Check whether this is the tightest window
   if temp.last − temp.first < distance
   then
      do (distance ← temp.last − temp.first
      window ← (temp.first, temp.last)
      return (distance, window
```

**Figure 4.3:** The distance class, only discussed members. This is a simplified picture, in reality specified distance inherits from k_distance, a similar class initiated with a $k$ that creates the thresholds $10 \times 2^x$, $x = 1, 2, ..., k$. 
4.3 Significance

In a pattern recognition context the term significance most commonly refers to the probability of observing the same pattern by chance, or in other words how likely the pattern is in a background model. The following is a discussion on how to evaluate and implement significance for the two motif representations discussed in section 3.1.

4.3.1 Regular Expressions

For regular expressions this project applies as significance a logarithm of the likelihood of hits in random sequences. The calculation estimates this by assuming independency of nucleotides in sequences, and using background likelihoods for the nucleotides. To ease the calculation of significance log2 of likelihoods are used. This allows for addition instead of multiplication. Ambiguities have to be treated as a single char of the pattern with regard to the significance. And the “any symbol” .’ has a likelihood of 1. This calculation results in the logarithm of the likelihood of matching a sequence in one position. $s_{pos}$ ($p-value$). To expand this into the significance one must evaluate the length of the pattern compared to the average length of sequences. This is computed as can be seen in equation 4.1 except in two special cases.

- Pattern is longer than the average sequence length
  - In this case the positional significance is used ($log2(p_{pos})$).
- The likelihood is so small as to create computational problems.
  - This happens at a likelihood of about $10^{-13}$.
  - In this case the calculation in equation 4.2 is used.

$$s_{seq} = log2(1 - (1 - p_{pos})^{l_{avg} - l})$$ (4.1)

$$s_{seq} = p_{pos} + log2(l_{avg} - l)$$ (4.2)

$s_{seq}$ Significance
$s_{pos}$ Possitional significance
$p_{pos}$ Possitional likelihood in background
$l_{avg}$ Average length of sequences
$l$ Pattern length

4.3.2 PWM

Significance for pwms are based on the same principle. The likelihood of matching random sequences, generated from the background. Still independence is assumed in the background. The background is essentially a Markov chain of order 0.

The main difference from the regular expression patterns, is that with regular expressions there is no need for a threshold value. Whether the reg.exp. hits a sequence is boolean. With PWMs however there is a score calculated for each position, and it therefore needs a threshold to define a hit. This threshold value is the score needed to achieve a low enough likelihood over the background, and is therefore defined by the significance or visa versa.

The most usually approach to setting these values is to specify the wanted significance (p-value). This achieves good control with the False Positives (FP) rate, since
all matches must have a small enough likelihood over the background to satisfy the p-value. In addition to this two other approaches has been implemented in this project. The second uses True Positives (TP) to set the threshold, and the third a balance between False Positives and False Negatives. All three are presently described.

**Significance set by False Positives**

The implementation of this solution is made by first defining the wanted granularity of scores. Here the default is to include one fractional digit. Than a buffer is created with a size equal to the difference between the highest possible score while traversing the PWM and the lowest (with the specified granularity). This buffer is indexed by scores and contains probabilities on background. Populating the buffer is done by traversing the positions in the PWM (columns). The very first column is just used as a plain index by adding the background likelihood of each nucleotide to the index specified by the PWM (see figure [4.4.3]). Thereafter the scores in the PWM can be viewed as offsets. Each index in the buffer containing a likelihood before this column is treated is moved according to the score and multiplied by the background probability for the current signal. The process is displayed in figure [4.4] The score buffer will always sum to 1, being probabilities. Conceptually this buffer can be thought of as the result of building a tree of the possible “paths” through the PWM, populating the nodes with the background probability. The score buffer would be the leaf nodes, indexed by score.

**Figure 4.4:** Visualisation of threshold calculation. Granularity kept at integers for simplicity

By adding the values of the buffer from right to left until one has achieved the target p-value, one can read the PWM threshold out of the index of the score buffer.
4.4. THE PARETO FRONT

CHAPTER 4. DESIGN AND IMPLEMENTATION

Significance by Balancing False Positives and False Negatives

In addition to the positional p-value, or the probability of a score equal or greater than $t$ when the sequence segment is generated by the background model, this approach needs to calculate the probability of observing a $t$ below given that the sequence segment was generated by the PWM. Call these values $\alpha(t)$ and $\beta(t)$ respectively. Then the threshold value $t$ and thereby the significance is defined by the formulae in equation (4.3). The variable $c$ in the equation allows for weighting the importance of the two error probabilities.

$$\alpha(t)c = \beta(t)$$

(4.3)

Implementation of this alternative is similar to what is displayed in figure 4.4. It needs another buffer for $\beta(t)$ which is populated with the likelihoods from the pwm instead of the background probabilities. These probabilities are calculated as in equation (3.1).

Significance by specifying True Positives

In this approach the support is specified, and the significance is set to allow this. The implementation is fairly simple. Sequences are scanned with each PWM and the highest score for each sequence recorded. Thereafter the scores are sorted and the $x$'th highest score is used as threshold for this PWM. Significance is then calculated by creating the same probability buffer, as with the two other approaches and adding the probabilities above the threshold.

4.4 The Pareto Front

Since the Support and Significance of Motifs constitute a Multiobjective Optimization Problem a Pareto Front is used for storing and comparing motifs. For single motifs this would be a vector of significance indexed by support. For this vector to have only elements which are Pareto optimal the elements in the vector have to be uniformly lower with higher significance. The difference is displayed in figure 4.5. This is done by applying a “smoothing”, resulting in each support inheriting the significance of any higher support with better significance. The storage for motifs in the search will henceforth be referred to as “pareto”, although it is a data structure consisting of multiple pareto fronts.

Figure 4.5: Visualisation of the effect of Pareto optimality. Here support plotted against significance.

Only the pareto for the target number of components is needed, but since the algorithm guarantees optimality with lower number of components aswell ,a front for each
number of components is kept. This leads to one Pareto front for each value of components. (see figure 4.6)

In addition - this project is also interested in distance. Therefore another dimension is introduced to the Pareto. This is the distance classes. Each distance class is defined by a threshold on distance, depending on what kind of restraint is used. This results in an internal structure like displayed in figure 4.7.

The Pareto is responsible for evaluating the motifs added. The search itself is dumb with respect to whether the motif is actually a part of the Pareto front. The reasoning behind this approach lies with keeping all the Pareto fronts below target number of components. When a motif is added, all the variants of the motif are checked against the contents of the Pareto. If any of them qualify - a new motif is created from the variant and inserted. (get_mav in figure 4.1) During the search there will be fewer of the variants, as the distance classes are pruned. (see sections 4.5 and 4.6).

The Pareto can produce a string representation of itself and an evaluation string of the “best” motifs for each distance class. Functionality to write the Pareto object to hard-drive for further post processing has been implemented but is not supported by the current stable version of Jython.
4.5 Composite Search

Motifs are combined in an implicit depth first tree through recursion, see algorithm 4.5.1 (Visualisation in figure 4.6) Every time a motif is added to a module and it is not caught by pruning it is passed to the pareto (see section 4.5). It is up to the pareto to check whether it is more significant that what currently is located there, and take appropriate action.

Since the motifs are pre-sorted on significance, it is likely that the early combinations will be the best. This effect is somewhat diminished by the distance restraints. Since the composite motifs must confer to different distance classes, it is not enough that they keep a high support and are significant, but they must also avoid spanning out of their distance class. This makes the pruning less effective (see: 4.6), but as distance restriction is one of the main objectives this is accepted.

Algorithm 4.5.1: COMPOSITE MOTIF SEARCH(pareto, numcomp)

procedure SEARCH(X)
  X.sort()
  c ← X.pop(0)
  BRANCH(c, X)

procedure BRANCH(c, X)
  if c.components < numcomp
    if PRUNE(c)
      then return (0)
      pareto.insert(c)
    for each motif ∈ X
      new_c ← c
      new_c.add(X.pop(0))
      do
        if PRUNE(c)
          then continue
          else BRANCH(new_c, X)
    else
      if not PRUNE(c)
        then pareto.insert(c)

4.6 Pruning

There are two values that can be used for pruning, support and significance. Leaving variants out of the picture, the pruning is presently described. With support pruning is straight forward. Since support cannot be gained by adding another motif, the present support is the maximum for the current combination. If this is lower then some user specified threshold the branch (of the search tree) can be safely pruned. With significance the picture is a bit more complex, since significance may indeed increase by adding motifs. For pruning to have any effect it must happen above the leafnodes, the higher up the more combinations can be left out of the search. To achieve this the potential significance of expanding the motif to the target number of component must
be evaluated, and compared to what is currently in the pareto. Given that the motifs are sorted in a descending fashion before the search starts a upper bound on the potential significance can be calculated as follows. Let $\text{sign}$ be the significance of the composite motif, $\text{last}$ be the significance of the last component added, $\text{components}$ be the target number of components, and $\text{cur}_\text{comp}$ be the current number in the composite. If

$$\text{sign} + \text{last} \times (\text{components} - \text{cur}_\text{comp}) < \text{pareto}$$ (4.4)

where $\text{pareto}$ pareto is the current value in the pareto front, at the appropriate significance, the motif can be safely pruned. It is impossible for the motif to attain greater significance then what is currently in the pareto front, at this level of support. Given "pareto optimality" (see figure 4.5) we now that we know that significance for lower values of support have to be equal or higher. This along with the fact that the motif cannot gain support guarantees that this composite can have no expansion that will be a part of the pareto front.

Taking variants back into account, the actual pruning mechanisms are much the same. However the bitsets for the variants, and therefore the support will differ. This is because not all sequences will have the combination of motifs within the tightest distance window. Therefore the pruning has to be performed on a pr. variant basis. In this project this is achieved in the following manner every time $\text{prune}$ is called in the search algorithm.

1. $\text{prune}$ is called on the composite motif
2. The composite calls $\text{calc_dist_class}$ on the distance object, retrieving a list of minimal distances for each sequence. (algorithm 4.2.1)
3. $\text{Check validity}$ is called on each of the composites variants, with the minimal distance list. This causes the variants to update themselves with new bitsets and significances. They also prune themselves against the pareto, using the approaches described above.

Thereby when the motif is passed to the pareto only valid variants that may achieve a place in the pareto front are present. When all the variants are pruned the entire branch can be quited. A visualisation of this can be seen in figure 4.6.

### 4.7 Evaluation of Modules

To evaluate the success of the algorithm in benchmarks one usually has to select only one module. To facilitate making this choice the composite motifs has a method for evaluating themselves. This is basically a binomial hypothesis test, where the null hypothesis is that the motif is generated by the background model. Equation 4.5 displays the actual formulae. It results in a one-tailed significance, the p-value of observed support, that can be used to compare the modules.

$$\sum_{x=s}^{s_{\text{max}}} \binom{s_{\text{max}}}{x} p^x (1 - p)^{s_{\text{max}} - x}$$ (4.5)

### 4.8 Choice of Language

The choice of languages are limited to the following.
4.8. CHOICE OF LANGUAGE CHAPTER 4. DESIGN AND IMPLEMENTATION

Figure 4.8: Visualisation of the search. Shows how Variants are pruned along the search

- Java
- C++
- Python
- Combination of the above

Reuseability
One important aspect of the implementation of this tool is the reusability of the sourcecode. Reusability is made a principle point to facilitate for further development by others than the current developer.

On this point, Python is an obvious candidate. Python is known to be an “easy to use, and understand” language. This because of its easily sloooping learning curve, and the way it promotes an “easy to read” coding style.

Another point is that furture work on this tool is probably to be made at NTNU. NTNU has in many ways choosen Java as a primary language, which should count in favour of Java.

Ease of development
To be able to produce as good a tool as possible in the allotted time, the ease of development is likely to have a major impact on the quality and diversity of the tool.

Also, unitests are preferred so that testing can be a part of the development process, and not something left to the end of the project that may be skipped. All three languages provide a unitesting framework, although python being an interpreted language lends itself easier to unitesting.

Efficiency
Efficiency will always be a major point in all tools such as this. A language unable to do the nessesary calculations efficiency must be dissqualified for choice of implementation language.

The Bitset
Most of the running time of this tool will be operation on bitsets. Therefore test have
been run on the efficiency of the different language candidates on execution times on
bitsets.

<table>
<thead>
<tr>
<th>Language</th>
<th>Seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>C++</td>
<td>0.17</td>
</tr>
<tr>
<td>Java</td>
<td>0.18</td>
</tr>
<tr>
<td>Python</td>
<td>1.31</td>
</tr>
<tr>
<td>Python with psyco</td>
<td>0.69</td>
</tr>
</tbody>
</table>

**Table 4.1:** The amount of time to do 1000 AND operations on 1000 bitsets 1000 times.

Appendix A.1 displays the results of these tests and describes how they where performed. (The code can be reviewed in A.2) Table 4.1 makes a summary of the results.

### 4.8.1 Conclusion

The implementation language decided upon is Jython. Thereby enabling the use of bitsets from Java while keeping the ease of development and potentially “easy to understand” source code from Python.

### 4.8.2 Language Limitations

The choice of Jython as implementation language has many benefits, but some drawbacks as well. This is mainly a result of Jython building on python 2.1, which is an edition of Python that can be said to be a bit “young”. For this reason a util module has been created to implement frequently used functions that are not available in Python 2.1, or that are special for this project. This also contributes to making the code elsewhere more readable, since some complexity may be hidden here.

Some functionality however is lost with the language choice. It is currently impossible to serialize the pareto front (including motifs, bitsets and sequence information) both with Javas ObjectOutputStream and Pythons Pickle, due to the mix of Java and Python objects. Since this is believed to change with future Jython versions, code to achieve this is included.
Chapter 5

Usage of the diplom tool

This chapter consists of an introduction to diplom in a practical sense.

5.1 Input

Diplom requires two input files. The motifs and the sequences respectively. The sequences must conform to FASTA format. With every second line being a sequence and the others the sequence name preceded by a “>”. An example of the format is displayed in listing 5.1.

Listing 5.1: Sequence File

>YKL016C
CAAAGACATAGTCGAACAAAAATATAATTAGGTGATTCTGCAATTAAGTGCACGAAAAATTTCTTTTAAATCCTAAGATTGTTCTTTTGCGTTACGTGTCAATATGAATATATAAACTTATATAGAAAAGTATTGTACTAGTAGAAATATGTTTTACACCTACGCTAAGCAGCAAGTGGTAATTGGTGTATCTTTTACTTTTTTTGGGGGCATCAAGACAAATATCCAATCAAATCGAAGAGAAAATAATTCTTGTAACCCGCTCATTA
GTCACGTGGTCATCACGTGAGCATAATATATCACGTGATATACACCCATACCTCGTT

>YBR248C
ATTAAAAAATCTAAACTGGATACTGCTACTTCAATAGCTGCCTCTTTTCTTTTAAAACCTGATTGAGTAGTCGTCGATATCAAAGGAATATCAACTTATGTATGTTTCGATGTCTGACTCTTTTCTCATGAATTTTTCATTTTTCATGATCACCTA

The motifs can be in one of two formats, regular expressions and PWMs. Motifs represented by regular expressions are much like the sequence file in that every second line is the name preceded by “>”, and the other the regular expression. Lines starting with # are considered to be comments and ignored. The line with the pattern must also contain the match- and positional information see listing 5.2. The different columns in the pattern line are as follows:

1. The total number of hits
2. Number of hits in distinct sequences
3. The pattern, only the ambiguities in braces and the wildcard “.” allowed.
4. The rest are pairs of sequence number and position in the sequence for each match.
5.2 Output

If the option “--outle <file>” is specified two files will be written. They are names <file>.log and <file>.eval. The logfile contains a textual representation of the pareto, a list of the motifs used, and a list of motif dropped as a result of some threshold if any. The evalfile contains details for the composite motif chosen by the formulae in equation 4.5. One motif is chosen for each distance class.

5.3 Commandline and Configuration

The commandline options are described in the manpage in appendix B and will be omitted here. The entire configuration has not been made available as options however, and the remaining are described here. The following parameters reside in conf.py in the system package.

- min_support [1]
  This is a lower threshold on accepted coverage.

- revcomp [0]
  Boolean - Whether the reverse complement should be considered. This is equivalent to searching the other strand, orientation the same with respect three-prime / five-prime

- rev [0]
  Boolean - search for the motif with opposite orientation, on the same strand.

- comp [0]
  Boolean - search the opposite strand, orientation opposite with respect to tree-prime / five-prime

- wide [0]
  Boolean - preserve motifs with significance within 0.1 of the pareto front.

The PWM motifs are alignment matrices on the format in listing 5.3 Although it expects alignment matrices float counts are accepted. The columns represent positions in the pattern, and the rows are A,C,G and T respectively.
• max_seq_hits [30]
  A threshold on the maximum number of hits in one sequence. This is an optimization to avoid too many permutations in the distance calculations.
Chapter 6

Results

The tool has been run on two different sets of datasets. For both sets only one composite motif is chosen from each distance class for evaluation. This is done as described in section 4.7. The datasets come with known sites and their positions, and python scripts for comparison to output.

Tests run using the “FPFN-approach” to setting significance and threshold gave no viable results, and none are therefore presented. Instead the more traditional approaches are used.

6.1 TransCompel Datasets

The first datasets are part of a benchmark created by members of the bioinformatics group at the Norwegian University of Science and Technology (NTNU), and stems from the TransCompel database. TransCompel is a specialised database of composite regulatory elements. That is a combination of two binding sites of two different transcription factors, which through this combination form a module with new regulatory properties.

The datasets have been created by retrieving from TransCompel all modules with matches in five or more promotor regions have been selected, along with the sequences in which they match. From this data three categories have been created. These are created by introducing noise in the form of additional matrices assumed not to be significant in the sequences. The categories are the percentage of matrices that should match the sequences. The percentages are set at 25, 50 and 100. A description of how the tests where run and the results themselves are presented in section 6.3.

6.2 Yeast Datasets

As with the TranCompel datasets, the yeast sets are provided by members of the bioinformatics group at NTNU. The sets consist of the match matrices and assorted decoys, their sequences and lists of known members for evaluation. The datasets stem from ChIP-CHIP experiments, where protein binding is recorded. This set has no positional information and is therefore evaluated on what motif presence alone. The results are presented in section 6.4.
6.3 TransCompel Results

A series of tests where run on the set with only match matrices, to get some idea of how well the matrices could be discerned from the background. The reason for this is to evaluate what the sequence likelihood has to be to predict matches in a reasonable amount of sequences in a positive set. The test series consisted of running the toll with input p-value (sequence likelihood) set to 0.05, 0.1, 0.2, 0.3, 0.4 and 0.5. The results are presented in figure 6.3 with an extra set labelled fasit that represent the number of sequences actually containing matches, which is also the total number of sequences.

![Graph showing the number of sequences with predicted match for different input p-values.](image)

**Figure 6.1:** Achieved support on a positive set, with different sequence likelihoods as input, and no restriction on distance.

As can be seen from the figure 6.3 the motifs differ only slightly from the background model. Specially the dataset *Ets-AML* in which there is no prediction until the sequence p-value is set to 0.5, which denotes a 50% chance of matching a pattern generated by the background.

With the transcompel datasets the evaluation is performed on a pr. nucleotide position basis. Figure 6.3 suggests the concept. The values nTP, nFP, nFN and nFP are added over the datasets and the statistics presented in chapter 3 are calculated and presented. Below each table of results over the combined dataset a barchart with the correlation coefficients of each single dataset is show for comparison. Please note that these charts differ in resolutions. A complete set of results is found in appendix C.
The test where also run with the different orientation options described in section 5.3 without any significant difference to the result, so those results have been omitted. The same is the case where the sequence files for all datasets were concatenated and used for background, as this had little or no effect on the probabilities in the background model.

### 6.3.1 100% sets

TransCompel results for TP significance approach

<table>
<thead>
<tr>
<th>D. Class</th>
<th>nSn</th>
<th>nPPV</th>
<th>nSP</th>
<th>nPC</th>
<th>nCC</th>
<th>nASP</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>0.151</td>
<td>0.026</td>
<td>0.954</td>
<td>0.023</td>
<td>0.044</td>
<td>0.089</td>
</tr>
<tr>
<td>75</td>
<td>0.065</td>
<td>0.137</td>
<td>0.997</td>
<td>0.046</td>
<td>0.09</td>
<td>0.101</td>
</tr>
<tr>
<td>150</td>
<td>0.116</td>
<td>0.105</td>
<td>0.992</td>
<td>0.058</td>
<td>0.103</td>
<td>0.111</td>
</tr>
<tr>
<td>225</td>
<td>0.109</td>
<td>0.075</td>
<td>0.989</td>
<td>0.046</td>
<td>0.081</td>
<td>0.092</td>
</tr>
</tbody>
</table>

Table 6.1: Results from likelihoods set to 0.8, 0.9 and 1.0

**Figure 6.2:** The position of the prediction and the known site displayed in relation to the sequence.

**Figure 6.3:** Correlation Coefficient for the Individual Datasets.
TransCompel results for FP significance approach

<table>
<thead>
<tr>
<th>Likelihood</th>
<th>D. Class</th>
<th>nSn</th>
<th>nPPV</th>
<th>nSP</th>
<th>nPC</th>
<th>nCC</th>
<th>nASP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2, 0.3, 0.4</td>
<td>None</td>
<td>0.284</td>
<td>0.014</td>
<td>0.82</td>
<td>0.014</td>
<td>0.025</td>
<td>0.149</td>
</tr>
<tr>
<td></td>
<td>75</td>
<td>0.051</td>
<td>0.253</td>
<td>0.999</td>
<td>0.044</td>
<td>0.11</td>
<td>0.152</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>0.199</td>
<td>0.428</td>
<td>0.998</td>
<td>0.157</td>
<td>0.288</td>
<td>0.314</td>
</tr>
<tr>
<td></td>
<td>225</td>
<td>0.244</td>
<td>0.314</td>
<td>0.995</td>
<td>0.159</td>
<td>0.271</td>
<td>0.279</td>
</tr>
<tr>
<td>0.4, 0.6, 0.8</td>
<td>None</td>
<td>0.299</td>
<td>0.063</td>
<td>0.962</td>
<td>0.055</td>
<td>0.122</td>
<td>0.181</td>
</tr>
<tr>
<td></td>
<td>75</td>
<td>0.162</td>
<td>0.308</td>
<td>0.997</td>
<td>0.119</td>
<td>0.219</td>
<td>0.235</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>0.168</td>
<td>0.205</td>
<td>0.994</td>
<td>0.102</td>
<td>0.179</td>
<td>0.186</td>
</tr>
<tr>
<td></td>
<td>225</td>
<td>0.168</td>
<td>0.156</td>
<td>0.992</td>
<td>0.088</td>
<td>0.155</td>
<td>0.162</td>
</tr>
</tbody>
</table>

Table 6.2: Results on 100% set, FP significance

Figure 6.4: Correlation Coefficients for likelihoods at 0.2, 0.3 and 0.4

Figure 6.5: Correlation Coefficient for likelihoods at 0.4, 0.6 and 0.8
6.3.2 50% sets

TransCompel results for TP significance approach

<table>
<thead>
<tr>
<th>D. Class</th>
<th>nSn</th>
<th>nPPV</th>
<th>nSP</th>
<th>nPC</th>
<th>nCC</th>
<th>nASP</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>0.216</td>
<td>0.033</td>
<td>0.948</td>
<td>0.029</td>
<td>0.065</td>
<td>0.124</td>
</tr>
<tr>
<td>75</td>
<td>0.12</td>
<td>0.185</td>
<td>0.996</td>
<td>0.079</td>
<td>0.144</td>
<td>0.153</td>
</tr>
<tr>
<td>150</td>
<td>0.105</td>
<td>0.069</td>
<td>0.989</td>
<td>0.044</td>
<td>0.076</td>
<td>0.087</td>
</tr>
<tr>
<td>225</td>
<td>0.099</td>
<td>0.051</td>
<td>0.985</td>
<td>0.035</td>
<td>0.06</td>
<td>0.075</td>
</tr>
</tbody>
</table>

Table 6.3: Results from likelihoods set to 0.8, 0.9 and 1.0

Figure 6.6: Correlation Coefficient for the Individual Datasets.
TransCompel results for FP significance approach

<table>
<thead>
<tr>
<th>Likelihoods</th>
<th>D. Class</th>
<th>nSn</th>
<th>nPPV</th>
<th>nSP</th>
<th>nPC</th>
<th>nCC</th>
<th>nASP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2, 0.3, 0.4</td>
<td>None</td>
<td>0.3</td>
<td>0.038</td>
<td>0.931</td>
<td>0.035</td>
<td>0.085</td>
<td>0.169</td>
</tr>
<tr>
<td></td>
<td>75</td>
<td>0.054</td>
<td>0.15</td>
<td>0.997</td>
<td>0.041</td>
<td>0.086</td>
<td>0.102</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>0.214</td>
<td>0.294</td>
<td>0.995</td>
<td>0.141</td>
<td>0.245</td>
<td>0.254</td>
</tr>
<tr>
<td></td>
<td>225</td>
<td>0.23</td>
<td>0.265</td>
<td>0.994</td>
<td>0.14</td>
<td>0.24</td>
<td>0.247</td>
</tr>
<tr>
<td>0.4, 0.6, 0.8</td>
<td>None</td>
<td>0.321</td>
<td>0.045</td>
<td>0.941</td>
<td>0.041</td>
<td>0.1</td>
<td>0.183</td>
</tr>
<tr>
<td></td>
<td>75</td>
<td>0.127</td>
<td>0.246</td>
<td>0.997</td>
<td>0.091</td>
<td>0.172</td>
<td>0.186</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>0.122</td>
<td>0.133</td>
<td>0.993</td>
<td>0.068</td>
<td>0.12</td>
<td>0.128</td>
</tr>
<tr>
<td></td>
<td>225</td>
<td>0.121</td>
<td>0.102</td>
<td>0.991</td>
<td>0.059</td>
<td>0.103</td>
<td>0.112</td>
</tr>
</tbody>
</table>

Table 6.4: Results on 50% set, FP significance

Figure 6.7: Correlation Coefficients for likelihoods at 0.2, 0.3 and 0.4

Figure 6.8: Correlation Coefficient for likelihoods at 0.4, 0.6 and 0.8
6.3.3 25% set

TransCompel results for TP significance approach

<table>
<thead>
<tr>
<th>D. Class</th>
<th>nSn</th>
<th>nPPV</th>
<th>nSP</th>
<th>nPC</th>
<th>nCC</th>
<th>nASP</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>0.424</td>
<td>0.038</td>
<td>0.91</td>
<td>0.036</td>
<td>0.105</td>
<td>0.231</td>
</tr>
<tr>
<td>75</td>
<td>0.131</td>
<td>0.178</td>
<td>0.995</td>
<td>0.082</td>
<td>0.147</td>
<td>0.155</td>
</tr>
<tr>
<td>150</td>
<td>0.116</td>
<td>0.069</td>
<td>0.987</td>
<td>0.045</td>
<td>0.08</td>
<td>0.092</td>
</tr>
<tr>
<td>225</td>
<td>0.107</td>
<td>0.044</td>
<td>0.982</td>
<td>0.032</td>
<td>0.058</td>
<td>0.076</td>
</tr>
</tbody>
</table>

Table 6.5: Results from likelihoods set to 0.8, 0.9 and 1.0

Figure 6.9: Correlation Coefficient for the Individual Datasets.
### 6.3. TRANSCOMPEL RESULTS

#### Chapter 6. Results

TransCompel results for FP significance approach

<table>
<thead>
<tr>
<th>Likelihoods</th>
<th>D. Class</th>
<th>nSn</th>
<th>nPPV</th>
<th>nSP</th>
<th>nPC</th>
<th>nCC</th>
<th>nASP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2, 0.3, 0.4</td>
<td>None</td>
<td>0.332</td>
<td>0.032</td>
<td>0.91</td>
<td>0.03</td>
<td>0.078</td>
<td>0.182</td>
</tr>
<tr>
<td></td>
<td>75</td>
<td>0.037</td>
<td>0.121</td>
<td>0.998</td>
<td>0.029</td>
<td>0.062</td>
<td>0.079</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>0.223</td>
<td>0.255</td>
<td>0.994</td>
<td>0.135</td>
<td>0.232</td>
<td>0.239</td>
</tr>
<tr>
<td></td>
<td>225</td>
<td>0.196</td>
<td>0.162</td>
<td>0.991</td>
<td>0.097</td>
<td>0.17</td>
<td>0.179</td>
</tr>
<tr>
<td>0.4, 0.6, 0.8</td>
<td>None</td>
<td>0.363</td>
<td>0.03</td>
<td>0.898</td>
<td>0.028</td>
<td>0.079</td>
<td>0.196</td>
</tr>
<tr>
<td></td>
<td>75</td>
<td>0.138</td>
<td>0.25</td>
<td>0.996</td>
<td>0.098</td>
<td>0.181</td>
<td>0.194</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>0.144</td>
<td>0.155</td>
<td>0.993</td>
<td>0.081</td>
<td>0.143</td>
<td>0.149</td>
</tr>
<tr>
<td></td>
<td>225</td>
<td>0.133</td>
<td>0.084</td>
<td>0.988</td>
<td>0.054</td>
<td>0.096</td>
<td>0.108</td>
</tr>
</tbody>
</table>

Table 6.6: Results on 25% set, FP significance

![Figure 6.10: Correlation Coefficients for likelihoods at 0.2, 0.3 and 0.4](image1)

![Figure 6.11: Correlation Coefficient for likelihoods at 0.4, 0.6 and 0.8](image2)
6.4 Yeast Results

In the results displayed the values sTP, sFP, sFN and sTN have been combined from all the datasets, and the evaluation values calculated from there as if it was one big dataset. This arrangement was suggested by Tompa et. al [28]. In the results with yeast the values are not connected to nucleotide positions but rather just the presence of the motif in the sequence.

Table Explanation

The first two columns in the result tables are the method used to set significance and the distance class or the size of the “distance restriction window”. The next six columns are the evaluation values presented in section 3.6. Some of the tables has an additional column labelled “N”. This column is the number of datasets where no composite motif could be discerned within the restrictions.

### Yeast results for likelihoods 0.05, 0.1, 0.2

<table>
<thead>
<tr>
<th>Sign</th>
<th>D. Class</th>
<th>sSn</th>
<th>sPPV</th>
<th>sSP</th>
<th>sPC</th>
<th>sCC</th>
<th>sASP</th>
<th>Sign</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>FP</td>
<td>None</td>
<td>0.364</td>
<td>0.45</td>
<td>0.954</td>
<td>0.252</td>
<td>0.35</td>
<td>0.407</td>
<td>8.297</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.32</td>
<td>0.36</td>
<td>0.945</td>
<td>0.204</td>
<td>0.28</td>
<td>0.34</td>
<td>8.97</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.316</td>
<td>0.392</td>
<td>0.951</td>
<td>0.212</td>
<td>0.295</td>
<td>0.354</td>
<td>8.993</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>0.392</td>
<td>0.482</td>
<td>0.957</td>
<td>0.276</td>
<td>0.384</td>
<td>0.437</td>
<td>8.471</td>
<td>0</td>
</tr>
<tr>
<td>TP</td>
<td>None</td>
<td>0.325</td>
<td>0.353</td>
<td>0.941</td>
<td>0.204</td>
<td>0.276</td>
<td>0.339</td>
<td>17.426</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.209</td>
<td>0.235</td>
<td>0.933</td>
<td>0.124</td>
<td>0.150</td>
<td>0.222</td>
<td>12.697</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.283</td>
<td>0.307</td>
<td>0.938</td>
<td>0.173</td>
<td>0.229</td>
<td>0.295</td>
<td>17.604</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>0.284</td>
<td>0.297</td>
<td>0.934</td>
<td>0.170</td>
<td>0.223</td>
<td>0.291</td>
<td>18.392</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 6.7: Results for the Yeast dataset (see section 6.4). The results displayed where achieved with likelihood set to 0.05, 0.1 and 0.2, and a maximum number of hits for one motif in one sequence of 10.

### Yeast results for likelihoods 0.2, 0.4, 0.6

<table>
<thead>
<tr>
<th>Sign</th>
<th>D. Class</th>
<th>sSn</th>
<th>sPPV</th>
<th>sSP</th>
<th>sPC</th>
<th>sCC</th>
<th>sASP</th>
<th>Sign</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>FP</td>
<td>None</td>
<td>0.319</td>
<td>0.362</td>
<td>0.944</td>
<td>0.204</td>
<td>0.279</td>
<td>0.341</td>
<td>4.071</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.199</td>
<td>0.214</td>
<td>0.928</td>
<td>0.115</td>
<td>0.131</td>
<td>0.207</td>
<td>4.917</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.241</td>
<td>0.269</td>
<td>0.935</td>
<td>0.145</td>
<td>0.185</td>
<td>0.255</td>
<td>4.435</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>0.29</td>
<td>0.335</td>
<td>0.942</td>
<td>0.184</td>
<td>0.248</td>
<td>0.312</td>
<td>4.314</td>
<td>0</td>
</tr>
<tr>
<td>TP</td>
<td>None</td>
<td>0.348</td>
<td>0.373</td>
<td>0.942</td>
<td>0.22</td>
<td>0.299</td>
<td>0.361</td>
<td>10.746</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.357</td>
<td>0.379</td>
<td>0.944</td>
<td>0.225</td>
<td>0.309</td>
<td>0.368</td>
<td>10.752</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.309</td>
<td>0.341</td>
<td>0.942</td>
<td>0.194</td>
<td>0.263</td>
<td>0.325</td>
<td>10.61</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>0.321</td>
<td>0.353</td>
<td>0.942</td>
<td>0.202</td>
<td>0.275</td>
<td>0.337</td>
<td>10.581</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 6.8: Results for the Yeast dataset (see section 6.4). The results displayed where achieved with likelihood set to 0.2, 0.4 and 0.6, and a maximum number of hits for one motif in one sequence of 10.
Yeast results for likelihoods 0.4, 0.6, 0.8

<table>
<thead>
<tr>
<th>Sign</th>
<th>D. Class</th>
<th>sSn</th>
<th>sPPV</th>
<th>sSP</th>
<th>sPC</th>
<th>sCC</th>
<th>sASP</th>
<th>Sign</th>
</tr>
</thead>
<tbody>
<tr>
<td>FP</td>
<td>None</td>
<td>0.344</td>
<td>0.373</td>
<td>0.942</td>
<td>0.218</td>
<td>0.297</td>
<td>0.359</td>
<td>2.567</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.14</td>
<td>0.15</td>
<td>0.926</td>
<td>0.078</td>
<td>0.069</td>
<td>0.145</td>
<td>2.73</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.224</td>
<td>0.243</td>
<td>0.932</td>
<td>0.132</td>
<td>0.162</td>
<td>0.233</td>
<td>2.609</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>0.22</td>
<td>0.237</td>
<td>0.93</td>
<td>0.129</td>
<td>0.155</td>
<td>0.229</td>
<td>2.282</td>
</tr>
<tr>
<td>TP</td>
<td>None</td>
<td>0.293</td>
<td>0.311</td>
<td>0.935</td>
<td>0.178</td>
<td>0.235</td>
<td>0.302</td>
<td>7.222</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.336</td>
<td>0.342</td>
<td>0.939</td>
<td>0.204</td>
<td>0.278</td>
<td>0.339</td>
<td>7.736</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.324</td>
<td>0.354</td>
<td>0.942</td>
<td>0.204</td>
<td>0.277</td>
<td>0.339</td>
<td>7.334</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>0.31</td>
<td>0.326</td>
<td>0.936</td>
<td>0.189</td>
<td>0.252</td>
<td>0.318</td>
<td>6.939</td>
</tr>
</tbody>
</table>

Table 6.9: likelihood 0.4, 0.6 and 0.8, maximum hits pr. sequence 10
Chapter 7

Discussion

7.1 TransCompel

As can be seen from figure 6.3 the motifs appear, to the algorithm, very similar to what can be generated by the background model. That is, the algorithm has difficulties finding the matches on single motifs. This can be caused by several factors. The PWMs used in the datasets are combinations of experimentally validated factors, and may be a bit too general to find good matches. Additionally the datasets allow overlaps, while the algorithm does not. This may cause combinations of two motifs to lose a significant part of the single motifs’ support. The literature argues that a higher order markov chain as background yields a positive effect, but the factor is not believed to be large.

Reviewing the actual results a fairly small but positive correlation coefficient is present throughout the combined datasets. Much of the reason for the correlation been low is believed to lie with the problem of attaining matches for the single motifs. Further evidence for this view can be gained from graphs in figures 6.3, 6.11. The results are based on combinations of datasets into what can be thought of as one big set. The graphs display the correlation for the individual datasets. It can be seen for the graphs that some datasets lack correlation entirely, as no modules were found for the set.

In the results with all approaches the highest correlation is found with restriction on distance. This is as expected in this dataset, since it consists of TFBS that are close together. The sensitivity, specificity and the positive predictive value suggest that this is due to an increase in true negatives and a decrease in false positives rather then more true positives. This does not however diminish the value of the restriction. A related effect can be seen from the graphs (figures 6.3, 6.11). In many cases the correlation is equal for several distance classes. The reason for this is that algorithm 4.2.1 chooses the least possible distance without regard for what the distance classes are. This is discussed further in section 7.3.

7.2 Yeast

The yeast datasets are based on biological ChIP-CHIP experiments only and the expectations to the results therefore must be moderate. This also excludes positional information, for this reason evaluation is based on presence alone. As can be seen in section 6.4 correlations are all positive, and generally above 0.15.
7.3. ALGORITHM

In this dataset the correlation is not consistently greater with distance restrictions. Studying the results separately however does reveal a trend. By looking at the results for “FP-significance” approach, it seems that when the input p-values are low (most reliable results) the correlation, specificity and sensitivity are highest when a distance restriction of 200 nucleotides is imposed. For the “TP-significance” approach the likelihood is the factor of sequences in which we want matches. For this approach the smallest “likelihoods” are too small as reflected in the significance, if compared to the significances of results in “FP-approach”. Keeping these considerations in mind it is plausible to assume that the data adhere to some constraint on distance.

7.3 Algorithm

As mentioned in section 7.1 the search effectively chooses the tightest possible window for each composite motif. This is a result of algorithm 4.2.1 which always returns the tightest possible parameters. The choice might be a cause for concern, since causes the chosen pattern to be the same for all distance classes in which it will fit. A plausible remedy might be to choose the pattern closest to the window restriction. This would in the worst case resolve the problem with no discrimination between the distance classes, and possibly enable use of the tool to evaluate distance restrictions based on the result. This could be done much in the same way as it currently is, by returning a list of tuples instead. The tuples would then consist of the widest window within the various distance classes. This would discriminate more between distance classes while still keeping the advantage of doing it once for all variants of the motif.
Bibliography


[34] V. Pareto. Cours deconomie politique. 1896.


[43] Øystein Lekang. Gige is gcmd expanded.
Appendix A

Efficiency Test on Bitsets

A.1 Results and Description

Tests where run on a Intel(R) Pentium(R) 4 CPU 3.00GHz running Debian GNU/Linux 2.4.27 for 689-smp. C++ compiler was run with the “-O3” optimizer, and one of the python code uses the “psyco” library as can be seen in the source code.

For the timing of the C++ code integers are used so the values are not inserted by the precompiler.

<table>
<thead>
<tr>
<th>Language</th>
<th>Compiler</th>
<th>Version</th>
<th>Time in Seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>C++</td>
<td>GCC/G++</td>
<td>3.3.6</td>
<td>0.17</td>
</tr>
<tr>
<td>Java</td>
<td>Blackdown</td>
<td>1.4.2-02</td>
<td>0.175</td>
</tr>
<tr>
<td>Python (psyco)</td>
<td>Python</td>
<td>2.3.5</td>
<td>0.69</td>
</tr>
<tr>
<td>Python</td>
<td>Python</td>
<td>2.3.5</td>
<td>1.31</td>
</tr>
</tbody>
</table>

Table A.1: Efficiency test of bitsets

A.2 Source Code

Listing A.1: Java Bitset Test

```java
import java.util.BitSet;
import java.util.Random;

class test{
    static int ANT_BITSET = 1000;
    static int STR_BITSET = 1000;
    static int ANT_GANGER_AND = 1000;
    static BitSet[] data;
```
public static void main(String[] args) {
    data = generate();
    tid(data);
}

static BitSet[] generate() {
    Random ran = new Random();
    BitSet[] return = new BitSet[ANT_BITSET];
    for (int j = 0; j < ANT_BITSET; j++) {
        BitSet x = new BitSet(STR_BITSET);
        for (int i = 0; i < STR_BITSET; i++) {
            if (ran.nextBoolean()) {
                x.set(i);
            }
        }
        return[j] = x;
    }
    return return;
}

static void tid(BitSet[] data) {
    long before, after;
    before = System.currentTimeMillis();
    for (int j = 0; j < ANT_JANGER_AND; j++) {
        for (int i = 0; i < ANT_BITSET; i++) {
            data[i].and(data[ANT_BITSET - i - 1]);
        }
    }
    after = System.currentTimeMillis();
    System.out.print("start: ");
    System.out.print(before);
    System.out.print(" slutt: ");
    System.out.print(after);
    System.out.print(" totalt: ");
    System.out.print(after - before);
    System.out.print(" sek: (\diff(millisecond)/1000) ");
    System.out.print((float)(after - before)/1000.0);
    System.out.println();
}
Listing A.2: C++ Bitset Test

```cpp
#include <vector>
#include <iostream>
#include <ctime>
#include <cstdlib>
#include <boost/dynamic_bitset.hpp>

using boost::dynamic_bitset;
using namespace std;

vector<dynamic_bitset<> > generate () {
    vector<dynamic_bitset<> > retu;
    for(int j=0; j < ANT_BITSET; j++){
        dynamic_bitset<> x(STR_BITSET);
        for(int i=0; i < STR_BITSET; i++){
            x[i] = abs(ceil(2*rand() / (RAND_MAX + 1.0)));
        }
        retu.push_back(x);
    }
    return retu;
}

void tid(vector<dynamic_bitset<> > data) {
    clock_t fortid , ettertid ;
    fortid = clock();
    for(int j = 0 ; j < ANT_GANGER_AND; j++){
        for(int i = 0; i < ANT_BITSET; i++) {
            data[i] = data[i] & data[ANT_BITSET - i - 1] ;
        }
    }
    ettertid = clock();
    cout << " start: " << fortid << " stopp: " << ettertid 
        << " totalt: " << ettertid - fortid
```
**A.2. SOURCE CODE**

**APPENDIX A. EFFICIENCY TEST ON BITSETS**

```c
51 << " sek: (diff/CLOCKS_PR_SEC) "
52 << (float) (etertid -fortid) / (3 * CLOCKS_PER_SEC) << '\n';
54 }
56 int main(int argc, char* argv[]) {
58 vector<dynamic_bitset <> > data = generate();
59 tid(data);
61 }
```

Note: With POSIX compliant compilers CLOCKS_PER_SEC is defined as 1000000.

---

**Listing A.3: Python Bitset Test**

```python
1 import Numeric
2 import random
3 import time
4 try:
5 try:
6     import psyco
7     psyco.profile()
8 except:
9     print 'Psyco not found, ignoring it'
10 ANT_BITSET = 1000
11 ANT_GANGER_AND = 1000
12 STR_BITSET = 1000
14 class bitset:
16     def __init__(self, liste):
17         self._bitset = Numeric.numarray(liste, Numeric.numarray.Bool)
19     def __getitem__(self, y):
20         return self._bitset[y]
22     def __setitem__(self, i, f):
23         self._bitset[i] = f
26     def AND(self, bitset):
27         if len(self._bitset) != len(bitset._bitset):
28             return -1
29     for i, f in enumerate(self._bitset):
30         pass
31     def main():
32         data = generate()
33         tid(data)
35     def generate():
36         return
```

---

52
```python
return = []
for f in range(ANT_BITSET):
    a = [random.randint(0,1) for f in range(STR_BITSET)]
    return.append(a)
return return

def tid(data):
i = 0
test = data[:]
before = time.clock()
while i < ANT_GANGER_AND:
    for i, f in enumerate(data):
        data[i] = [j.__and__(data[STR_BITSET -1 -i][k]) \n                   for k, j in enumerate(f)]
i += 1
after = time.clock()
print "start:" + str(before) + " stopp:" + str(after) \n      + " total:" + str(after - before)
if __name__ == "__main__":
    main()
```
Appendix B

Manual page

diplom(1)

NAME
diplom − program to discover sets of motif combinations with distance restrictions.

SYNOPSIS
diplom [options] sequencefile patternfile

DESCRIPTION
This manual page documents briefly the diplom program.

diplom is based on "General Composite Motif Discovery", and is a tool to find sets of motifs that occur together in sequences of a positive data set. Hence bettering the significance and sensitivity of the motif. It also finds the best within every given distance window, if specified.

OPTIONS
This program follows the usual GNU command line syntax, with long options starting with two dashes ('-').

All options must be stated before the arguments. A summary of options is included below.

−h, −−help
Show summary of options.

−b, −−background <file>
Use <file> to generate background model. If not specified the background model is created from the sequence file.

−d, −−distances d1,d2,d3,...
Comma separated lists of the distance restrictions you want diplom to run. The default is one distance class, half the size of the average sequence length.

−s, −−signtype FP | TP | FP,FN
The mode of significance calculation wanted.
FP - Default, likelihoods treated as target p-values.
TP - likelihoods treated as coverage. Values below 1 are treated as factors of sequence count.
FP,FN - likelihoods are treated as a factor C in the equation "ErrorI * C = ErrorII"

−l, −−likelihoods l0,l1,l2,..
Comma separated list of likelihoods. The evaluation of the values depend on the signtype. Default is that they are interpreted as target p-values.

−o, −−outfile <basename>
This specifies the basename of output files. There will be two. One log file and one eval file. The log file contains an ascii representation of the pareto, and the eval file contains the information on the best composite for each distance class.

−c, −−components <number>
The target number of components in a composite motif.

−v
Verbose output to screen. Yields more information on the progress of the algorithm.

AUTHOR
diplom was written by Øystein Lekang, for his masters degree.

This manual page was written by Øystein Lekang <lekang@idi.ntnu.no>.

Jun 20, 2006
Appendix C

Complete TransCompel Results

-s TP -l 0.8,0.9,1.0

### Listing C.1: 100% Combined

**distance: 0**

<table>
<thead>
<tr>
<th></th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
<th>nASP</th>
<th>nCC</th>
<th>nPC</th>
<th>nPPV</th>
<th>nSP</th>
<th>nSn</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1208</td>
<td>8067</td>
<td>167137</td>
<td>215</td>
<td>0.089</td>
<td>0.044</td>
<td>0.023</td>
<td>0.015</td>
<td>0.967</td>
<td>0.077</td>
</tr>
</tbody>
</table>

**distance: 75**

<table>
<thead>
<tr>
<th></th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
<th>nASP</th>
<th>nCC</th>
<th>nPC</th>
<th>nPPV</th>
<th>nSP</th>
<th>nSn</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1278</td>
<td>559</td>
<td>166474</td>
<td>89</td>
<td>0.101</td>
<td>0.095</td>
<td>0.046</td>
<td>0.026</td>
<td>0.997</td>
<td>0.065</td>
</tr>
</tbody>
</table>

**distance: 150**

<table>
<thead>
<tr>
<th></th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
<th>nASP</th>
<th>nCC</th>
<th>nPC</th>
<th>nPPV</th>
<th>nSP</th>
<th>nSn</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1599</td>
<td>1331</td>
<td>165711</td>
<td>157</td>
<td>0.111</td>
<td>0.103</td>
<td>0.058</td>
<td>0.023</td>
<td>0.992</td>
<td>0.116</td>
</tr>
</tbody>
</table>

**distance: 225**

<table>
<thead>
<tr>
<th></th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
<th>nASP</th>
<th>nCC</th>
<th>nPC</th>
<th>nPPV</th>
<th>nSP</th>
<th>nSn</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1290</td>
<td>1953</td>
<td>173226</td>
<td>158</td>
<td>0.092</td>
<td>0.081</td>
<td>0.046</td>
<td>0.026</td>
<td>0.998</td>
<td>0.109</td>
</tr>
</tbody>
</table>

### Listing C.2: 100% Distance 75

**AP1**

<table>
<thead>
<tr>
<th></th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
<th>nASP</th>
<th>nCC</th>
<th>nPC</th>
<th>nPPV</th>
<th>nSP</th>
<th>nSn</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>144</td>
<td>161</td>
<td>18060</td>
<td>9</td>
<td>0.056</td>
<td>0.047</td>
<td>0.029</td>
<td>0.015</td>
<td>0.987</td>
<td>0.077</td>
</tr>
</tbody>
</table>

**NFAT**

<table>
<thead>
<tr>
<th></th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
<th>nASP</th>
<th>nCC</th>
<th>nPC</th>
<th>nPPV</th>
<th>nSP</th>
<th>nSn</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>388</td>
<td>0</td>
<td>8297</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.998</td>
<td>0.0</td>
</tr>
</tbody>
</table>

**Ebox**

<table>
<thead>
<tr>
<th></th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
<th>nASP</th>
<th>nCC</th>
<th>nPC</th>
<th>nPPV</th>
<th>nSP</th>
<th>nSn</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>134</td>
<td>20</td>
<td>13006</td>
<td>7</td>
<td>0.155</td>
<td>0.109</td>
<td>0.043</td>
<td>0.012</td>
<td>0.998</td>
<td>0.05</td>
</tr>
</tbody>
</table>

**IRF**

<table>
<thead>
<tr>
<th></th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
<th>nASP</th>
<th>nCC</th>
<th>nPC</th>
<th>nPPV</th>
<th>nSP</th>
<th>nSn</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>64</td>
<td>260</td>
<td>14914</td>
<td>50</td>
<td>0.078</td>
<td>0.058</td>
<td>0.027</td>
<td>0.033</td>
<td>0.983</td>
<td>0.123</td>
</tr>
</tbody>
</table>

**Sp1**

<table>
<thead>
<tr>
<th></th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
<th>nASP</th>
<th>nCC</th>
<th>nPC</th>
<th>nPPV</th>
<th>nSP</th>
<th>nSn</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>235</td>
<td>244</td>
<td>13869</td>
<td>50</td>
<td>0.173</td>
<td>0.158</td>
<td>0.099</td>
<td>0.173</td>
<td>0.983</td>
<td>0.175</td>
</tr>
</tbody>
</table>

### Listing C.3: 100% Distance 150

**AP1**

<table>
<thead>
<tr>
<th></th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
<th>nASP</th>
<th>nCC</th>
<th>nPC</th>
<th>nPPV</th>
<th>nSP</th>
<th>nSn</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>108</td>
<td>765</td>
<td>17492</td>
<td>9</td>
<td>0.045</td>
<td>0.014</td>
<td>0.002</td>
<td>0.009</td>
<td>0.958</td>
<td>0.077</td>
</tr>
</tbody>
</table>

**NFAT**

<table>
<thead>
<tr>
<th></th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
<th>nASP</th>
<th>nCC</th>
<th>nPC</th>
<th>nPPV</th>
<th>nSP</th>
<th>nSn</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>385</td>
<td>0</td>
<td>8297</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.998</td>
<td>0.0</td>
</tr>
</tbody>
</table>

**Ebox**

<table>
<thead>
<tr>
<th></th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
<th>nASP</th>
<th>nCC</th>
<th>nPC</th>
<th>nPPV</th>
<th>nSP</th>
<th>nSn</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>134</td>
<td>20</td>
<td>13006</td>
<td>7</td>
<td>0.155</td>
<td>0.109</td>
<td>0.043</td>
<td>0.012</td>
<td>0.998</td>
<td>0.05</td>
</tr>
</tbody>
</table>

**IRF**

<table>
<thead>
<tr>
<th></th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
<th>nASP</th>
<th>nCC</th>
<th>nPC</th>
<th>nPPV</th>
<th>nSP</th>
<th>nSn</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>64</td>
<td>260</td>
<td>14914</td>
<td>50</td>
<td>0.078</td>
<td>0.058</td>
<td>0.027</td>
<td>0.033</td>
<td>0.983</td>
<td>0.123</td>
</tr>
</tbody>
</table>

**Sp1**

<table>
<thead>
<tr>
<th></th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
<th>nASP</th>
<th>nCC</th>
<th>nPC</th>
<th>nPPV</th>
<th>nSP</th>
<th>nSn</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>235</td>
<td>244</td>
<td>13869</td>
<td>50</td>
<td>0.173</td>
<td>0.158</td>
<td>0.099</td>
<td>0.173</td>
<td>0.983</td>
<td>0.175</td>
</tr>
</tbody>
</table>

### Listing C.4: 100% Distance 225

**AP1**

<table>
<thead>
<tr>
<th></th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
<th>nASP</th>
<th>nCC</th>
<th>nPC</th>
<th>nPPV</th>
<th>nSP</th>
<th>nSn</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>108</td>
<td>765</td>
<td>17492</td>
<td>9</td>
<td>0.045</td>
<td>0.014</td>
<td>0.002</td>
<td>0.009</td>
<td>0.958</td>
<td>0.077</td>
</tr>
</tbody>
</table>

**NFAT**

<table>
<thead>
<tr>
<th></th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
<th>nASP</th>
<th>nCC</th>
<th>nPC</th>
<th>nPPV</th>
<th>nSP</th>
<th>nSn</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>385</td>
<td>0</td>
<td>8297</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.998</td>
<td>0.0</td>
</tr>
</tbody>
</table>

**Ebox**

<table>
<thead>
<tr>
<th></th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
<th>nASP</th>
<th>nCC</th>
<th>nPC</th>
<th>nPPV</th>
<th>nSP</th>
<th>nSn</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>134</td>
<td>20</td>
<td>13006</td>
<td>7</td>
<td>0.155</td>
<td>0.109</td>
<td>0.043</td>
<td>0.012</td>
<td>0.998</td>
<td>0.05</td>
</tr>
</tbody>
</table>

**IRF**

<table>
<thead>
<tr>
<th></th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
<th>nASP</th>
<th>nCC</th>
<th>nPC</th>
<th>nPPV</th>
<th>nSP</th>
<th>nSn</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>64</td>
<td>260</td>
<td>14914</td>
<td>50</td>
<td>0.078</td>
<td>0.058</td>
<td>0.027</td>
<td>0.033</td>
<td>0.983</td>
<td>0.123</td>
</tr>
</tbody>
</table>

**Sp1**

<table>
<thead>
<tr>
<th></th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
<th>nASP</th>
<th>nCC</th>
<th>nPC</th>
<th>nPPV</th>
<th>nSP</th>
<th>nSn</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>235</td>
<td>244</td>
<td>13869</td>
<td>50</td>
<td>0.173</td>
<td>0.158</td>
<td>0.099</td>
<td>0.173</td>
<td>0.983</td>
<td>0.175</td>
</tr>
</tbody>
</table>
### Listing C.5: 100% Combined

<table>
<thead>
<tr>
<th>distance</th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>153</td>
<td>106</td>
<td>18115</td>
<td>0</td>
</tr>
<tr>
<td>75</td>
<td>153</td>
<td>152</td>
<td>17440</td>
<td>0</td>
</tr>
<tr>
<td>225</td>
<td>153</td>
<td>153</td>
<td>17605</td>
<td>313</td>
</tr>
<tr>
<td>750</td>
<td>153</td>
<td>153</td>
<td>17635</td>
<td>199</td>
</tr>
</tbody>
</table>

### Listing C.6: 100% Distance 75

<table>
<thead>
<tr>
<th>API-NEAT</th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>153</td>
<td>0</td>
<td>0</td>
<td>17440</td>
<td>0</td>
</tr>
<tr>
<td>153</td>
<td>0</td>
<td>152</td>
<td>18115</td>
<td>0</td>
</tr>
<tr>
<td>153</td>
<td>153</td>
<td>153</td>
<td>17440</td>
<td>0</td>
</tr>
<tr>
<td>153</td>
<td>153</td>
<td>153</td>
<td>17605</td>
<td>313</td>
</tr>
</tbody>
</table>

### Listing C.7: 100% Distance 150

<table>
<thead>
<tr>
<th>API-NEAT</th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>106</td>
<td>0</td>
<td>0</td>
<td>17440</td>
<td>0</td>
</tr>
<tr>
<td>152</td>
<td>0</td>
<td>0</td>
<td>17440</td>
<td>0</td>
</tr>
<tr>
<td>153</td>
<td>153</td>
<td>153</td>
<td>17440</td>
<td>0</td>
</tr>
</tbody>
</table>

### Listing C.8: 100% Distance 225

<table>
<thead>
<tr>
<th>API-NEAT</th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>153</td>
<td>0</td>
<td>0</td>
<td>17440</td>
<td>0</td>
</tr>
<tr>
<td>153</td>
<td>0</td>
<td>152</td>
<td>18115</td>
<td>0</td>
</tr>
<tr>
<td>153</td>
<td>153</td>
<td>153</td>
<td>17440</td>
<td>0</td>
</tr>
</tbody>
</table>

### Listing C.9: 100% Combined

<table>
<thead>
<tr>
<th>distance</th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>153</td>
<td>106</td>
<td>18115</td>
<td>0</td>
</tr>
<tr>
<td>75</td>
<td>153</td>
<td>152</td>
<td>17440</td>
<td>0</td>
</tr>
<tr>
<td>225</td>
<td>153</td>
<td>153</td>
<td>17605</td>
<td>313</td>
</tr>
<tr>
<td>750</td>
<td>153</td>
<td>153</td>
<td>17635</td>
<td>199</td>
</tr>
</tbody>
</table>

### Listing C.10: 100% Distance 75

<table>
<thead>
<tr>
<th>API-NEAT</th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>153</td>
<td>0</td>
<td>0</td>
<td>17440</td>
<td>0</td>
</tr>
<tr>
<td>153</td>
<td>0</td>
<td>152</td>
<td>18115</td>
<td>0</td>
</tr>
<tr>
<td>153</td>
<td>153</td>
<td>153</td>
<td>17440</td>
<td>0</td>
</tr>
</tbody>
</table>

### Listing C.11: 100% Distance 150

<table>
<thead>
<tr>
<th>API-NEAT</th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>106</td>
<td>0</td>
<td>0</td>
<td>17440</td>
<td>0</td>
</tr>
<tr>
<td>152</td>
<td>0</td>
<td>0</td>
<td>17440</td>
<td>0</td>
</tr>
<tr>
<td>153</td>
<td>153</td>
<td>153</td>
<td>17440</td>
<td>0</td>
</tr>
</tbody>
</table>

### Listing C.12: 100% Distance 225

<table>
<thead>
<tr>
<th>API-NEAT</th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>153</td>
<td>0</td>
<td>0</td>
<td>17440</td>
<td>0</td>
</tr>
<tr>
<td>153</td>
<td>0</td>
<td>152</td>
<td>18115</td>
<td>0</td>
</tr>
<tr>
<td>153</td>
<td>153</td>
<td>153</td>
<td>17440</td>
<td>0</td>
</tr>
</tbody>
</table>

### Listing C.13: 100% Distance 750

<table>
<thead>
<tr>
<th>API-NEAT</th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>153</td>
<td>0</td>
<td>0</td>
<td>17440</td>
<td>0</td>
</tr>
<tr>
<td>153</td>
<td>0</td>
<td>152</td>
<td>18115</td>
<td>0</td>
</tr>
<tr>
<td>153</td>
<td>153</td>
<td>153</td>
<td>17440</td>
<td>0</td>
</tr>
</tbody>
</table>

### Listing C.14: 100% Distance 150

<table>
<thead>
<tr>
<th>API-NEAT</th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>106</td>
<td>0</td>
<td>0</td>
<td>17440</td>
<td>0</td>
</tr>
<tr>
<td>152</td>
<td>0</td>
<td>0</td>
<td>17440</td>
<td>0</td>
</tr>
<tr>
<td>153</td>
<td>153</td>
<td>153</td>
<td>17440</td>
<td>0</td>
</tr>
</tbody>
</table>
APPENDIX C. COMPLETE TRANSCOMPIL RESULTS

Listing C.11: 100% Distance 150

<table>
<thead>
<tr>
<th>Gene</th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
<th>ASP</th>
<th>CC</th>
<th>PC</th>
<th>PPV</th>
<th>SP</th>
<th>Sn</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP1</td>
<td>153</td>
<td>321</td>
<td>17994</td>
<td>0</td>
<td>0.0</td>
<td>-</td>
<td>-</td>
<td>0.0</td>
<td>0.982</td>
<td>0.0</td>
</tr>
<tr>
<td>AP1</td>
<td>420</td>
<td>92</td>
<td>91040</td>
<td>54</td>
<td>0.242</td>
<td>-</td>
<td>-</td>
<td>0.099</td>
<td>0.113</td>
<td>0.0</td>
</tr>
<tr>
<td>NFκB</td>
<td>87</td>
<td>79</td>
<td>12949</td>
<td>52</td>
<td>0.386</td>
<td>-</td>
<td>-</td>
<td>0.916</td>
<td>0.378</td>
<td>0.239</td>
</tr>
<tr>
<td>Ets</td>
<td>96</td>
<td>151</td>
<td>13544</td>
<td>9</td>
<td>0.056</td>
<td>0.994</td>
<td>0.665</td>
<td>0.949</td>
<td>0.095</td>
<td>0.0</td>
</tr>
<tr>
<td>IRF3</td>
<td>175</td>
<td>114</td>
<td>9688</td>
<td>79</td>
<td>0.358</td>
<td>0.984</td>
<td>0.834</td>
<td>0.089</td>
<td>0.313</td>
<td>0.213</td>
</tr>
<tr>
<td>Sp1</td>
<td>235</td>
<td>97</td>
<td>14016</td>
<td>50</td>
<td>0.254</td>
<td>0.254</td>
<td>0.993</td>
<td>0.54</td>
<td>0.0</td>
<td>0.175</td>
</tr>
</tbody>
</table>

Listing C.12: 100% Distance 225

<table>
<thead>
<tr>
<th>Gene</th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
<th>ASP</th>
<th>CC</th>
<th>PC</th>
<th>PPV</th>
<th>SP</th>
<th>Sn</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP1</td>
<td>153</td>
<td>702</td>
<td>17519</td>
<td>0</td>
<td>0.0</td>
<td>-</td>
<td>-</td>
<td>0.018</td>
<td>0.961</td>
<td>0.0</td>
</tr>
<tr>
<td>AP1</td>
<td>420</td>
<td>92</td>
<td>91040</td>
<td>56</td>
<td>0.242</td>
<td>0.252</td>
<td>0.094</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>NFκB</td>
<td>87</td>
<td>79</td>
<td>12949</td>
<td>52</td>
<td>0.386</td>
<td>0.979</td>
<td>0.665</td>
<td>0.949</td>
<td>0.095</td>
<td>0.0</td>
</tr>
<tr>
<td>Ets</td>
<td>96</td>
<td>151</td>
<td>13544</td>
<td>9</td>
<td>0.056</td>
<td>0.994</td>
<td>0.665</td>
<td>0.949</td>
<td>0.095</td>
<td>0.0</td>
</tr>
<tr>
<td>IRF3</td>
<td>175</td>
<td>114</td>
<td>9688</td>
<td>79</td>
<td>0.358</td>
<td>0.984</td>
<td>0.834</td>
<td>0.089</td>
<td>0.313</td>
<td>0.213</td>
</tr>
<tr>
<td>Sp1</td>
<td>235</td>
<td>97</td>
<td>14016</td>
<td>50</td>
<td>0.254</td>
<td>0.254</td>
<td>0.993</td>
<td>0.54</td>
<td>0.0</td>
<td>0.175</td>
</tr>
</tbody>
</table>

Listing C.13: 50% Combined

<table>
<thead>
<tr>
<th>Gene</th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
<th>ASP</th>
<th>CC</th>
<th>PC</th>
<th>PPV</th>
<th>SP</th>
<th>Sn</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP1</td>
<td>153</td>
<td>321</td>
<td>17994</td>
<td>0</td>
<td>0.0</td>
<td>-</td>
<td>-</td>
<td>0.018</td>
<td>0.961</td>
<td>0.0</td>
</tr>
<tr>
<td>AP1</td>
<td>420</td>
<td>92</td>
<td>91040</td>
<td>56</td>
<td>0.242</td>
<td>0.252</td>
<td>0.094</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>NFκB</td>
<td>87</td>
<td>79</td>
<td>12949</td>
<td>52</td>
<td>0.386</td>
<td>0.979</td>
<td>0.665</td>
<td>0.949</td>
<td>0.095</td>
<td>0.0</td>
</tr>
<tr>
<td>Ets</td>
<td>96</td>
<td>151</td>
<td>13544</td>
<td>9</td>
<td>0.056</td>
<td>0.994</td>
<td>0.665</td>
<td>0.949</td>
<td>0.095</td>
<td>0.0</td>
</tr>
<tr>
<td>IRF3</td>
<td>175</td>
<td>114</td>
<td>9688</td>
<td>79</td>
<td>0.358</td>
<td>0.984</td>
<td>0.834</td>
<td>0.089</td>
<td>0.313</td>
<td>0.213</td>
</tr>
<tr>
<td>Sp1</td>
<td>235</td>
<td>97</td>
<td>14016</td>
<td>50</td>
<td>0.254</td>
<td>0.254</td>
<td>0.993</td>
<td>0.54</td>
<td>0.0</td>
<td>0.175</td>
</tr>
</tbody>
</table>

Listing C.14: 50% Distance 75

<table>
<thead>
<tr>
<th>Gene</th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
<th>ASP</th>
<th>CC</th>
<th>PC</th>
<th>PPV</th>
<th>SP</th>
<th>Sn</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP1</td>
<td>153</td>
<td>702</td>
<td>17519</td>
<td>0</td>
<td>0.0</td>
<td>-</td>
<td>-</td>
<td>0.018</td>
<td>0.961</td>
<td>0.0</td>
</tr>
<tr>
<td>AP1</td>
<td>420</td>
<td>92</td>
<td>91040</td>
<td>56</td>
<td>0.242</td>
<td>0.252</td>
<td>0.094</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>NFκB</td>
<td>87</td>
<td>79</td>
<td>12949</td>
<td>52</td>
<td>0.386</td>
<td>0.979</td>
<td>0.665</td>
<td>0.949</td>
<td>0.095</td>
<td>0.0</td>
</tr>
<tr>
<td>Ets</td>
<td>96</td>
<td>151</td>
<td>13544</td>
<td>9</td>
<td>0.056</td>
<td>0.994</td>
<td>0.665</td>
<td>0.949</td>
<td>0.095</td>
<td>0.0</td>
</tr>
<tr>
<td>IRF3</td>
<td>175</td>
<td>114</td>
<td>9688</td>
<td>79</td>
<td>0.358</td>
<td>0.984</td>
<td>0.834</td>
<td>0.089</td>
<td>0.313</td>
<td>0.213</td>
</tr>
<tr>
<td>Sp1</td>
<td>235</td>
<td>97</td>
<td>14016</td>
<td>50</td>
<td>0.254</td>
<td>0.254</td>
<td>0.993</td>
<td>0.54</td>
<td>0.0</td>
<td>0.175</td>
</tr>
</tbody>
</table>

Listing C.15: 50% Distance 150

<table>
<thead>
<tr>
<th>Gene</th>
<th>nFN</th>
<th>nFP</th>
<th>nTN</th>
<th>nTP</th>
<th>ASP</th>
<th>CC</th>
<th>PC</th>
<th>PPV</th>
<th>SP</th>
<th>Sn</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP1</td>
<td>153</td>
<td>702</td>
<td>17519</td>
<td>0</td>
<td>0.0</td>
<td>-</td>
<td>-</td>
<td>0.018</td>
<td>0.961</td>
<td>0.0</td>
</tr>
<tr>
<td>AP1</td>
<td>420</td>
<td>92</td>
<td>91040</td>
<td>56</td>
<td>0.242</td>
<td>0.252</td>
<td>0.094</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>NFκB</td>
<td>87</td>
<td>79</td>
<td>12949</td>
<td>52</td>
<td>0.386</td>
<td>0.979</td>
<td>0.665</td>
<td>0.949</td>
<td>0.095</td>
<td>0.0</td>
</tr>
<tr>
<td>Ets</td>
<td>96</td>
<td>151</td>
<td>13544</td>
<td>9</td>
<td>0.056</td>
<td>0.994</td>
<td>0.665</td>
<td>0.949</td>
<td>0.095</td>
<td>0.0</td>
</tr>
<tr>
<td>IRF3</td>
<td>175</td>
<td>114</td>
<td>9688</td>
<td>79</td>
<td>0.358</td>
<td>0.984</td>
<td>0.834</td>
<td>0.089</td>
<td>0.313</td>
<td>0.213</td>
</tr>
<tr>
<td>Sp1</td>
<td>235</td>
<td>97</td>
<td>14016</td>
<td>50</td>
<td>0.254</td>
<td>0.254</td>
<td>0.993</td>
<td>0.54</td>
<td>0.0</td>
<td>0.175</td>
</tr>
</tbody>
</table>

- TP - 0.8, 0.9, 1.0
## Listing C.16: 50% Distance 225

<table>
<thead>
<tr>
<th>Gene</th>
<th>AP1</th>
<th>NFκB</th>
<th>ETS</th>
<th>IRF</th>
<th>NFAT</th>
<th>Ets</th>
<th>NFkappaB</th>
<th>IFN</th>
<th>PPV</th>
<th>SP</th>
<th>Sn</th>
</tr>
</thead>
<tbody>
<tr>
<td>nFN</td>
<td>168</td>
<td>95</td>
<td>82</td>
<td>95</td>
<td>126</td>
<td>1017</td>
<td>1380</td>
<td>1224</td>
<td>570</td>
<td>539</td>
<td>60</td>
</tr>
<tr>
<td>nFP</td>
<td>330</td>
<td>40</td>
<td>480</td>
<td>95</td>
<td>1017</td>
<td>251</td>
<td>17425</td>
<td>145</td>
<td>145</td>
<td>674</td>
<td>60</td>
</tr>
<tr>
<td>nTN</td>
<td>17876</td>
<td>13646</td>
<td>15661</td>
<td>13940</td>
<td>164825</td>
<td>1923</td>
<td>174221</td>
<td>13040</td>
<td>493</td>
<td>13040</td>
<td>17425</td>
</tr>
<tr>
<td>nTP</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>480</td>
<td>230</td>
<td>79</td>
<td>585</td>
<td>585</td>
<td>585</td>
<td>585</td>
</tr>
</tbody>
</table>

### -s FP -10,2.0,3,0,4

## Listing C.17: 50% Combined

<table>
<thead>
<tr>
<th>Gene</th>
<th>AP1</th>
<th>NFκB</th>
<th>ETS</th>
<th>IRF</th>
<th>NFAT</th>
<th>Ets</th>
<th>NFkappaB</th>
<th>IFN</th>
<th>PPV</th>
<th>SP</th>
<th>Sn</th>
</tr>
</thead>
<tbody>
<tr>
<td>nFN</td>
<td>168</td>
<td>95</td>
<td>82</td>
<td>95</td>
<td>126</td>
<td>1017</td>
<td>1380</td>
<td>1224</td>
<td>570</td>
<td>539</td>
<td>60</td>
</tr>
<tr>
<td>nFP</td>
<td>330</td>
<td>40</td>
<td>480</td>
<td>95</td>
<td>1017</td>
<td>251</td>
<td>17425</td>
<td>145</td>
<td>145</td>
<td>674</td>
<td>60</td>
</tr>
<tr>
<td>nTN</td>
<td>17876</td>
<td>13646</td>
<td>15661</td>
<td>13940</td>
<td>164825</td>
<td>1923</td>
<td>174221</td>
<td>13040</td>
<td>493</td>
<td>13040</td>
<td>17425</td>
</tr>
<tr>
<td>nTP</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>480</td>
<td>230</td>
<td>79</td>
<td>585</td>
<td>585</td>
<td>585</td>
<td>585</td>
</tr>
</tbody>
</table>

### -s FP -10,2.0,3,0,4

## Listing C.18: 50% Distance 75

<table>
<thead>
<tr>
<th>Gene</th>
<th>AP1</th>
<th>NFκB</th>
<th>ETS</th>
<th>IRF</th>
<th>NFAT</th>
<th>Ets</th>
<th>NFkappaB</th>
<th>IFN</th>
<th>PPV</th>
<th>SP</th>
<th>Sn</th>
</tr>
</thead>
<tbody>
<tr>
<td>nFN</td>
<td>168</td>
<td>95</td>
<td>82</td>
<td>95</td>
<td>126</td>
<td>1017</td>
<td>1380</td>
<td>1224</td>
<td>570</td>
<td>539</td>
<td>60</td>
</tr>
<tr>
<td>nFP</td>
<td>330</td>
<td>40</td>
<td>480</td>
<td>95</td>
<td>1017</td>
<td>251</td>
<td>17425</td>
<td>145</td>
<td>145</td>
<td>674</td>
<td>60</td>
</tr>
<tr>
<td>nTN</td>
<td>17876</td>
<td>13646</td>
<td>15661</td>
<td>13940</td>
<td>164825</td>
<td>1923</td>
<td>174221</td>
<td>13040</td>
<td>493</td>
<td>13040</td>
<td>17425</td>
</tr>
<tr>
<td>nTP</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>480</td>
<td>230</td>
<td>79</td>
<td>585</td>
<td>585</td>
<td>585</td>
<td>585</td>
</tr>
</tbody>
</table>

### -s FP -10,2.0,3,0,4

## Listing C.19: 50% Distance 150

<table>
<thead>
<tr>
<th>Gene</th>
<th>AP1</th>
<th>NFκB</th>
<th>ETS</th>
<th>IRF</th>
<th>NFAT</th>
<th>Ets</th>
<th>NFkappaB</th>
<th>IFN</th>
<th>PPV</th>
<th>SP</th>
<th>Sn</th>
</tr>
</thead>
<tbody>
<tr>
<td>nFN</td>
<td>168</td>
<td>95</td>
<td>82</td>
<td>95</td>
<td>126</td>
<td>1017</td>
<td>1380</td>
<td>1224</td>
<td>570</td>
<td>539</td>
<td>60</td>
</tr>
<tr>
<td>nFP</td>
<td>330</td>
<td>40</td>
<td>480</td>
<td>95</td>
<td>1017</td>
<td>251</td>
<td>17425</td>
<td>145</td>
<td>145</td>
<td>674</td>
<td>60</td>
</tr>
<tr>
<td>nTN</td>
<td>17876</td>
<td>13646</td>
<td>15661</td>
<td>13940</td>
<td>164825</td>
<td>1923</td>
<td>174221</td>
<td>13040</td>
<td>493</td>
<td>13040</td>
<td>17425</td>
</tr>
<tr>
<td>nTP</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>480</td>
<td>230</td>
<td>79</td>
<td>585</td>
<td>585</td>
<td>585</td>
<td>585</td>
</tr>
</tbody>
</table>

### -s FP -10,2.0,3,0,4

## Listing C.20: 50% Distance 225

<table>
<thead>
<tr>
<th>Gene</th>
<th>AP1</th>
<th>NFκB</th>
<th>ETS</th>
<th>IRF</th>
<th>NFAT</th>
<th>Ets</th>
<th>NFkappaB</th>
<th>IFN</th>
<th>PPV</th>
<th>SP</th>
<th>Sn</th>
</tr>
</thead>
<tbody>
<tr>
<td>nFN</td>
<td>168</td>
<td>95</td>
<td>82</td>
<td>95</td>
<td>126</td>
<td>1017</td>
<td>1380</td>
<td>1224</td>
<td>570</td>
<td>539</td>
<td>60</td>
</tr>
<tr>
<td>nFP</td>
<td>330</td>
<td>40</td>
<td>480</td>
<td>95</td>
<td>1017</td>
<td>251</td>
<td>17425</td>
<td>145</td>
<td>145</td>
<td>674</td>
<td>60</td>
</tr>
<tr>
<td>nTN</td>
<td>17876</td>
<td>13646</td>
<td>15661</td>
<td>13940</td>
<td>164825</td>
<td>1923</td>
<td>174221</td>
<td>13040</td>
<td>493</td>
<td>13040</td>
<td>17425</td>
</tr>
<tr>
<td>nTP</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>480</td>
<td>230</td>
<td>79</td>
<td>585</td>
<td>585</td>
<td>585</td>
<td>585</td>
</tr>
</tbody>
</table>

### -s FP -10,2.0,3,0,4

## Listing C.21: 50% Combined

<table>
<thead>
<tr>
<th>Gene</th>
<th>AP1</th>
<th>NFκB</th>
<th>ETS</th>
<th>IRF</th>
<th>NFAT</th>
<th>Ets</th>
<th>NFkappaB</th>
<th>IFN</th>
<th>PPV</th>
<th>SP</th>
<th>Sn</th>
</tr>
</thead>
<tbody>
<tr>
<td>nFN</td>
<td>168</td>
<td>95</td>
<td>82</td>
<td>95</td>
<td>126</td>
<td>1017</td>
<td>1380</td>
<td>1224</td>
<td>570</td>
<td>539</td>
<td>60</td>
</tr>
<tr>
<td>nFP</td>
<td>330</td>
<td>40</td>
<td>480</td>
<td>95</td>
<td>1017</td>
<td>251</td>
<td>17425</td>
<td>145</td>
<td>145</td>
<td>674</td>
<td>60</td>
</tr>
<tr>
<td>nTN</td>
<td>17876</td>
<td>13646</td>
<td>15661</td>
<td>13940</td>
<td>164825</td>
<td>1923</td>
<td>174221</td>
<td>13040</td>
<td>493</td>
<td>13040</td>
<td>17425</td>
</tr>
<tr>
<td>nTP</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>480</td>
<td>230</td>
<td>79</td>
<td>585</td>
<td>585</td>
<td>585</td>
<td>585</td>
</tr>
</tbody>
</table>

### -s FP -10,2.0,3,0,4
### APPENDIX C. COMPLETE TRANSCOMPEL RESULTS

#### Listing C.22: 50% Distance 75

<table>
<thead>
<tr>
<th>AP1</th>
<th>nFN : 1285 nFP : 1157 nTN : 174007 nTP : 178</th>
<th>nPPV : 0.246 nSP : 0.997 nSn : 0.127</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP1</td>
<td>nFN : 1305 nFP : 1168 nTN : 173757 nTP : 179</td>
<td>nASP : 0.128 nCC : 0.12 nAPC : 0.068</td>
</tr>
<tr>
<td>AP1</td>
<td>nFN : 0 nFP : 0 nTN : 99999 nTP : 0</td>
<td>nPPV : 0.133 nSP : 0.993 nSn : 0.122</td>
</tr>
</tbody>
</table>

#### Listing C.23: 50% Distance 150

<table>
<thead>
<tr>
<th>AP1-APAT</th>
<th>nFN : 1305 nFP : 1168 nTN : 173757 nTP : 179</th>
<th>nASP : 0.128 nCC : 0.12 nAPC : 0.068</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP1-APAT</td>
<td>nFN : 0 nFP : 0 nTN : 99999 nTP : 0</td>
<td>nPPV : 0.133 nSP : 0.993 nSn : 0.122</td>
</tr>
</tbody>
</table>

#### Listing C.24: 50% Distance 225

<table>
<thead>
<tr>
<th>AP1-APAT</th>
<th>nFN : 1305 nFP : 1168 nTN : 173757 nTP : 179</th>
<th>nASP : 0.128 nCC : 0.12 nAPC : 0.068</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP1-APAT</td>
<td>nFN : 0 nFP : 0 nTN : 99999 nTP : 0</td>
<td>nPPV : 0.133 nSP : 0.993 nSn : 0.122</td>
</tr>
</tbody>
</table>

#### Listing C.25: 25% Combined

<table>
<thead>
<tr>
<th>AP1-APAT</th>
<th>nFN : 1305 nFP : 1168 nTN : 173757 nTP : 179</th>
<th>nASP : 0.128 nCC : 0.12 nAPC : 0.068</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP1-APAT</td>
<td>nFN : 0 nFP : 0 nTN : 99999 nTP : 0</td>
<td>nPPV : 0.133 nSP : 0.993 nSn : 0.122</td>
</tr>
</tbody>
</table>

#### Listing C.26: 25% Distance 75

<table>
<thead>
<tr>
<th>AP1-APAT</th>
<th>nFN : 1305 nFP : 1168 nTN : 173757 nTP : 179</th>
<th>nASP : 0.128 nCC : 0.12 nAPC : 0.068</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP1-APAT</td>
<td>nFN : 0 nFP : 0 nTN : 99999 nTP : 0</td>
<td>nPPV : 0.133 nSP : 0.993 nSn : 0.122</td>
</tr>
</tbody>
</table>

---

**s TP -10.8,0.9,1.0**
Listing C.27: 25% Distance 150

API–NFAT
aFP 106 aFP 996 aTN 15861 aTP 9
aASP 0.046 aCC 0.02 aSFC 0.013
aPPV 0.015 aSFP 0.967 aSN 0.077

API–NFkB
aSN 430 aSN 117 aTN 9105 aTP 0
aSFP 0.0 aCC –0.003 aSFC 0.0
aPPV 0.0 aSFP 0.999 aSN 0.0

Eho–Ets
aSN 93 aSN 183 aTN 12845 aTP 48
aSFP 0.974 aCC 0.558 aSFC 0.348
aPPV 0.268 aSFP 0.966 aSN 0.34

Ets–AP1
aSN 65 aSN 194 aTN 15541 aTP 30
aSFP 0.24 aCC 0.22 aSFC 0.12
aPPV 0.163 aSFP 0.999 aSN 0.316

NFAT–NFkB
aSN 218 aSN 288 aTN 9494 aTP 39
aSFP 0.136 aCC 0.109 aSFC 0.072
aPPV 0.119 aSFP 0.971 aSN 0.152

NFkB–BMEV
aSN 54 aSN 428 aTN 14746 aTP 39
aSFP 0.309 aCC 0.263 aSFC 0.078
aPPV 0.084 aSFP 0.972 aSN 0.534

Sp1–Ets
aSN 260 aSN 475 aTN 15663 aTP 0
aSFP 0.0 aCC –0.025 aSFC 0.0
aPPV 0.0 aSFP 0.966 aSN 0.0

Listing C.28: 25% Distance 225

API–NFAT
aSN 106 aSN 169 aTN 17892 aTP 9
aSFP 0.045 aCC 0.014 aSFC 0.01
aPPV 0.012 aSFP 0.958 aSN 0.077

API–NFkB
aSN 398 aSN 479 aTN 90715 aTP 22
aSFP 0.048 aCC 0.043 aSFC 0.024
aPPV 0.044 aSFP 0.995 aSN 0.052

Eho–Ets
aSN 93 aSN 183 aTN 12845 aTP 48
aSFP 0.274 aCC 0.256 aSFC 0.148
aPPV 0.208 aSFP 0.966 aSN 0.34

Ets–AP1
aSN 67 aSN 631 aTN 15974 aTP 28
aSFP 0.169 aCC 0.097 aSFC 0.039
aPPV 0.043 aSFP 0.955 aSN 0.285

NFAT–NFkB
aSN 218 aSN 288 aTN 9494 aTP 39
aSFP 0.136 aCC 0.109 aSFC 0.072
aPPV 0.119 aSFP 0.971 aSN 0.152

NFkB–BMEV
aSN 53 aSN 349 aTN 14825 aTP 0
aSFP 0.0 aCC –0.001 aSFC 0.0
aPPV 0.0 aSFP 0.977 aSN 0.0

Sp1–Ets
aSN 260 aSN 475 aTN 15663 aTP 0
aSFP 0.0 aCC –0.025 aSFC 0.0
aPPV 0.0 aSFP 0.966 aSN 0.0

-s FP -l 0.2,0.3,0.4

Listing C.29: 25% Combined

distance: 0
aSN 1046 aSN 15818 aTN 199242 aTP 521
aSFP 0.02 aCC 0.078 aSFC 0.03
aPPV 0.032 aSFP 0.91 aSN 0.532

distance: 75
aSN 1192 aSN 429 aTN 174607 aTP 59
aSFP 0.079 aCC 0.062 aSFC 0.029
aPPV 0.121 aSFP 0.998 aSN 0.037

distance: 150
aSN 1234 aSN 1034 aTN 174005 aTP 354
aSFP 0.329 aCC 0.232 aSFC 0.135
aPPV 0.255 aSFP 0.994 aSN 0.223

Listing C.30: 25% Distance 75

API–NFAT
aSN 168 aSN 48 aTN 18158 aTP 0
aSFP 0.0 aCC –0.005 aSFC 0.0
aPPV 0.0 aSFP 0.997 aSN 0.0

API–NFkB
aSN 480 aSN 40 aTN 91092 aTP 0
aSFP 0.0 aCC –0.002 aSFC 0.0
aPPV 0.0 aSFP 1.0 aSN 0.0

Eho–Ets
aSN 62 aSN 71 aTN 12955 aTP 59
aSFP 0.456 aCC 0.43 aSFC 0.278
aPPV 0.454 aSFP 0.995 aSN 0.418

Ets–AP1
aSN 99 aSN 47 aTN 13668 aTP 0
aSFP 0.0 aCC –0.005 aSFC 0.0
aPPV 0.0 aSFP 0.997 aSN 0.0

NFAT–NFkB
aSN 257 aSN 52 aTN 9730 aTP 0
aSFP 0.0 aCC –0.002 aSFC 0.0
aPPV 0.0 aSFP 0.995 aSN 0.0

NFkB–BMEV
aSN 73 aSN 99 aTN 15075 aTP 0
aSFP 0.0 aCC –0.006 aSFC 0.0
aPPV 0.0 aSFP 0.993 aSN 0.0

Sp1–Ets
aSN 577 aSN 72 aTN 13949 aTP 0
aSFP 0.0 aCC –0.002 aSFC 0.0
aPPV 0.0 aSFP 0.995 aSN 0.0

Listing C.31: 25% Distance 150

API–NFAT
aSN 168 aSN 242 aTN 17064 aTP 0
aSFP 0.0 aCC –0.001 aSFC 0.0
aPPV 0.0 aSFP 0.967 aSN 0.0

API–NFkB
aSN 480 aSN 40 aTN 91092 aTP 0
aSFP 0.0 aCC –0.002 aSFC 0.0
aPPV 0.0 aSFP 1.0 aSN 0.0

Eho–Ets
aSN 82 aSN 71 aTN 12955 aTP 59
aSFP 0.456 aCC 0.43 aSFC 0.278
aPPV 0.454 aSFP 0.995 aSN 0.418

Ets–AP1
aSN 99 aSN 47 aTN 13668 aTP 0
aSFP 0.0 aCC –0.005 aSFC 0.0
aPPV 0.0 aSFP 0.997 aSN 0.0

NFAT–NFkB
aSN 214 aSN 269 aTN 9777 aTP 43
aSFP 0.17 aCC 0.149 aSFC 0.093
aPPV 0.173 aSFP 0.979 aSN 0.167

NFkB–BMEV
aSN 65 aSN 143 aTN 15031 aTP 5
aSFP 0.042 aCC 0.04 aSFC 0.057
aPPV 0.033 aSFP 0.991 aSN 0.11

Sp1–Ets
aSN 130 aSN 286 aTN 13738 aTP 244
aSFP 0.558 aCC 0.534 aSFC 0.37
aPPV 0.46 aSFP 0.98 aSN 0.652

Listing C.32: 25% Distance 225

API–NFAT
aSN 168 aSN 350 aTN 17081 aTP 0
aSFP 0.0 aCC –0.003 aSFC 0.0
aPPV 0.0 aSFP 0.982 aSN 0.0

API–NFkB
aSN 480 aSN 40 aTN 91092 aTP 0
aSFP 0.0 aCC –0.002 aSFC 0.0
aPPV 0.0 aSFP 1.0 aSN 0.0

Eho–Ets
APPENDIX C. COMPLETE TRANSCOMPEL RESULTS

<table>
<thead>
<tr>
<th>AP1</th>
<th>AP1</th>
<th>AP1</th>
<th>AP1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ets</td>
<td>Ets</td>
<td>Ets</td>
<td>Ets</td>
</tr>
<tr>
<td>AML</td>
<td>AML</td>
<td>AML</td>
<td>AML</td>
</tr>
<tr>
<td>IRF</td>
<td>IRF</td>
<td>IRF</td>
<td>IRF</td>
</tr>
<tr>
<td>NFkB</td>
<td>NFkB</td>
<td>NFkB</td>
<td>NFkB</td>
</tr>
<tr>
<td>HMGIY</td>
<td>HMGIY</td>
<td>HMGIY</td>
<td>HMGIY</td>
</tr>
<tr>
<td>Sp1</td>
<td>Sp1</td>
<td>Sp1</td>
<td>Sp1</td>
</tr>
<tr>
<td>Ets</td>
<td>Ets</td>
<td>Ets</td>
<td>Ets</td>
</tr>
<tr>
<td>AML</td>
<td>AML</td>
<td>AML</td>
<td>AML</td>
</tr>
<tr>
<td>IRF</td>
<td>IRF</td>
<td>IRF</td>
<td>IRF</td>
</tr>
<tr>
<td>NFkB</td>
<td>NFkB</td>
<td>NFkB</td>
<td>NFkB</td>
</tr>
<tr>
<td>HMGIY</td>
<td>HMGIY</td>
<td>HMGIY</td>
<td>HMGIY</td>
</tr>
<tr>
<td>Sp1</td>
<td>Sp1</td>
<td>Sp1</td>
<td>Sp1</td>
</tr>
</tbody>
</table>