Master's Thesis

Using the Multiple Instance Learning framework to address differential regulation

Thesis Committee:
Prof.Dr.Ir. Marcel J.T. Reinders
Dr. David M.J. Tax
Dr. Michael T. Emmerich
Ir. Veronika Cheplygina
MSc. Thies Gehrmann

Author	**Dimitrios Palachanis**
Email	d.palachanis@gmail.com
Student number	**TU Delft**: 4195558 **Leiden University**: 0448850
Thesis supervisors	Dr. David M.J. Tax Dr. Michael T. Emmerich MSc. Thies Gehrmann
Date	May 28, 2014

TU Delft

Preface

The current document is the end result of the research done on the use of Multiple Instance Learning (MIL) for the answer of a biological question. The document consists of a paper, supplementary material and a work log document which describes in detail the work done throughout the thesis project.

The paper provides an introduction to the biological question and the use of MIL to answer it, and outlines the developed methods and results. Supplementary material is included which provides further explanations on the data sources used and additional figures that support some decisions that were made. The work document is a log of the work that was done in chronological order, omitting the last months, as all the focus was put into the writing of the paper.

This research was done in the Pattern Recognition and Bioinformatics group of the Intelligent Systems department in the faculty of Electrical Engineering, Mathematics and Computer Science at the Delft University of Technology under the supervision of Dr.ir. D.M.J. Tax, MSc Thies Gehrmann and Dr.rer.nat. M.T.M. Emmerich[1]. The thesis project was started in July 2013 and will be defended on May 28^{th} 2014 in Delft and May 30^{th} 2014 in LIACS.

[1]Dr. M.T.M. Emmerich is from the Leiden Institute of Advanced Computer Science

TU Delft Master's Thesis

May 28th 2014
Pages 1– 12

Using the Multiple Instance Learning framework to address differential regulation

Dimitrios Palachanis*

Pattern Recognition and Bioinformatics group† Delft University of Technology, Delft, The Netherlands

Defended on May 28th 2014

ABSTRACT

Cell differentiation is a natural process occurring in all higher organisms, since the early fetal stage of life. It is, also, a part of disease – such as cancer – as the cell cycle becomes deregulated and cells behave differently compared to healthy ones. Differentiation occurs although the genome of all cells is identical across all cell types of the same organism. The motivation behind the current work is to understand why this happens.

Cells differentiate because of different gene expression patterns. The genomic features close or around a gene determine its expression. One of these genomic features is the binding of Transcription Factors (TFs), which are proteins that bind in the promoter region of genes and are responsible for their (non-) expression. Other genomic features influence the binding of TFs close to genes, such as the accessibility of DNA, the levels of DNA methylation or the modification of histones.

The purpose of this study is to identify the genomic features that influence the binding of the TFs that are responsible for gene expression. Normal classification cannot express that multiple TFs need to bind in a gene's promoter region for it to be expressed and the number of TFs varies among genes. The TF labels are also unknown, meaning that it is not known which TF, or TFs, is/are responsible for gene expression.

For these reasons, this problem – and the data – fits the Multiple Instance Learning (MIL) framework. A method is formulated, where a gene is treated as a bag and all the TF binding sites are instances. The results are promising, as TFs that were selected as important for gene expression were found to be so in a biological example.

Contact: d.palachanis@gmail.com

1 INTRODUCTION

All cell types of an organism are differentiated from stem cells, to fulfil their different functions. In cancer, something similar happens, as a cancerous cell will differentiate from the healthy ones, become deregulated and start multiplying rapidly. Since the genome of all cells is the same, ignoring somatic mutations, it is logical to question what happened during differentiation, which results in cancer in one cell type, but not in other(s).

Stemming from biological knowledge is the fact that cell differentiation occurs due to differences in gene expression. Some genes are expressed in one cell type but not in others and vice versa, thus producing different proteins and making the cells have different functions.

The various gene expression patterns are a result of differences in genomic properties. These properties include: (*a*) accessibility of DNA, (*b*) binding of proteins to special DNA regions upstream or downstream of genes (enhancers/ silencers), (*c*) methylation of DNA, (*d*) binding of TFs in the promoter region of genes, (*e*) histone modifications, among others.

As can be seen in Fig. 1, the genomic properties can influence gene expression in many ways. The openness of chromatin makes the genome more accessible to binding proteins. In general, tightly compacted chromatin makes the genes inactive (Fig. 1a). TFs that bind to the promoter region of a gene, may activate it or render it inert (Fig. 1b). There is also a synergistic effect, as two, or more, TFs may be required for a gene to be expressed or not. A variable number of TFs is needed per gene to activate or deactivate it. Chromatin is an octamer, a protein complex consisting of 8 proteins, called histones. The histones have "tails" that protrude to the outside of the complex. These tails can be chemically modified (methylation, acetylation, phosphorylation) and some of the modifications have been associated with gene expression or inactivation. Fig. 1c depicts the N-tail of histone H3 in gray. Specific modifications of Lysine (K) and Serine (S) of this tail have been associated to gene silencing or expression. Finally, methylation of the DNA sequence is generally associated with gene silencing, as it inhibits TFs from binding (Fig. 1d).

This study focuses on the binding of TFs in the promoter region of genes. We wish to answer the biological question of which features of the genome are responsible for TF binding, thus making genes being expressed in one cell type, but not in others.

Every TF binds on a particular piece of DNA sequence, that is a TF binding site (TFBS), and that region is conserved across cell types. It can be considered as an unambiguous mapping of a TF on the genome. The number of TFBSs – and the corresponding TFs – varies per gene. For example, one gene may require two TFs to initiate transcription, while another may need to recruit only one or more than two.

The TFBSs can be described with their genomic features in a particular cell type. Focusing on the rest of the genomic properties in Fig. 1, each TFBS can be characterized with how accessible the DNA is at that location, with the levels of methylation and with nearby histone modifications.

If enough information is contained in these features, one would be able to classify and obtain the TFBSs responsible for the differences in gene expression. From the meaningfully classified TFBSs, the differences in the genomic properties can be obtained and be linked back to, or explain, gene expression patterns.

With a pattern recognition approach, one would try to predict gene expression. To achieve that, a dataset would be constructed by describing each gene with features and then a classifier would be fitted to some example genes and their desired expression labels. Unfortunately, this standard approach is not directly applicable,

*to whom correspondence should be addressed

†Pattern Recognition and Bioinformatics: http://prb.tudelft.nl

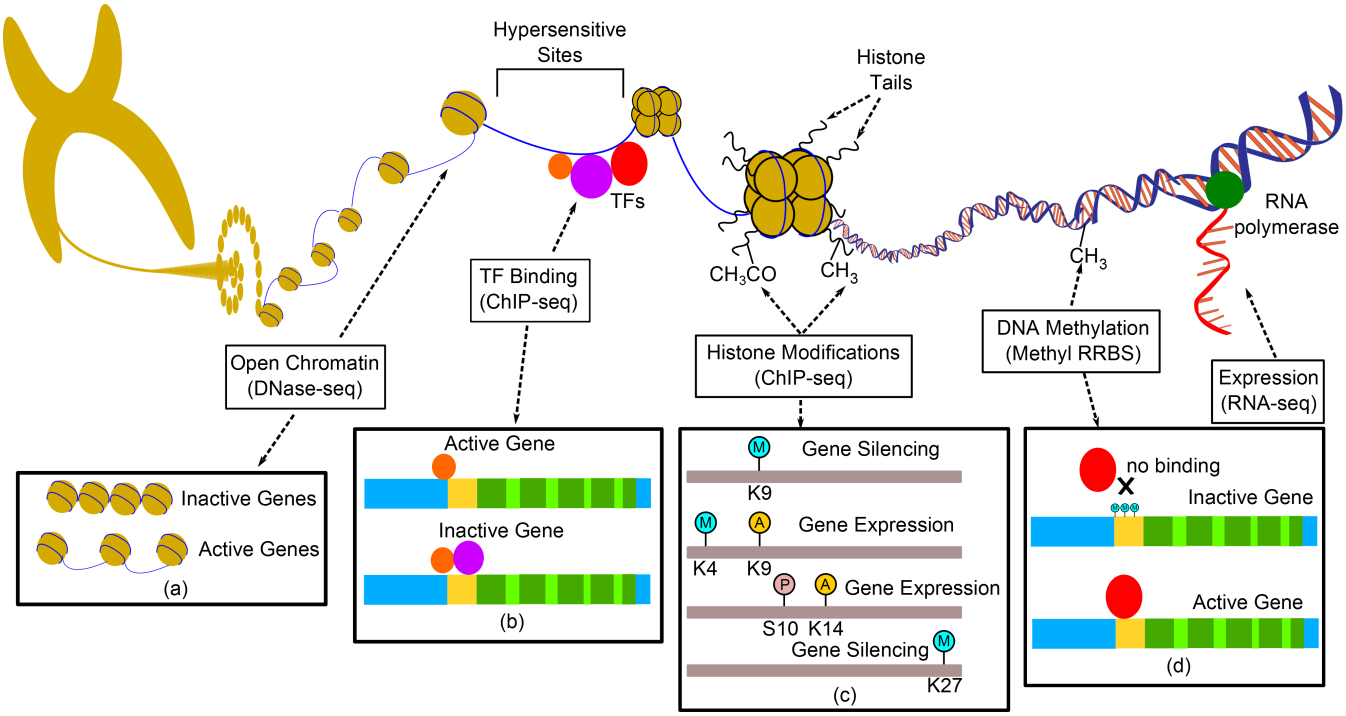


Fig. 1: **Effects of genomic features on gene expression: a**: Openness of DNA makes genes more accessible, while closeness makes them inactive. **b**: TF binding can aid or inhibit gene expression. The number of TFs that bind per gene is variable. **c**: Modifications of the N-tail of histone H3 and their association to gene expression. *M, A, P* stand for methylation, acetylation and phosphorylation respectively. *K, S* are the peptides Lysine and Serine, respectively. The numbers next to the peptides are the positions on the tail peptide sequence. **d**: An example of DNA methylation where a TF is inhibited from binding, thus making a gene inactive. The upper part of the figure is reproduced from https://genome.ucsc.edu/ ENCODE/aboutScaleup.html, ©Darryl Leja (NHGRI). Part **(d)** is reproduced from Fig. 4-44B of the Molecular Biology of the Cell, 5^{th} ed., p.226

because it assumes that there is a fixed set of characteristic gene features that can predict the gene activation well. Here, on the other hand, gene expression is based on a variable number of TFs. To overcome this limitation, a more flexible representation of a gene in terms of TFs is needed.

This more flexible representation is given by Multiple Instance Learning (MIL) (Dietterich, 1997), introduced in the context of drug activity prediction. In this framework, an object is represented by a collection of feature vectors – called a ”bag” of ”instances” –, instead of a single feature vector. In this way, a gene can be characterized by a set of feature vectors, each one describing a TFBS or, subsequently, a TF. When a MIL classifier is trained, it is able to predict from a bag of instances one label for the bag. Furthermore, some MIL classifiers are even able to extract/ highlight one single instance that is the most informative for the prediction of the bag label, also called a ”concept”. This offers, then, the additional possibility to interpret the classification result. For the prediction of gene expression, it may highlight the TF that is responsible for the activation of most genes. From there on, the genomic properties that cause this TF to be significant can be found.

Two scenarios are considered in this study. (*a*) The first is to identify TFs that are responsible for gene expression in one cell type, as can be seen in Fig. 2. This scenario is more meaningful for subsets of genes that may be involved in a particular function, as finding TFs that are meaningful ”globally”, for all genes in a particular cell type, is not deemed to have much biological merit. The genomic features of the most interesting TFs in this scenario can be linked to the particular biological function that the subset of genes is involved in. (*b*) The second scenario addresses differential regulation in multiple cell types. For a certain pattern of expression, genes are labelled appropriately and the most interesting TFBSs are found. Thus, the TFs responsible for the differential regulation of genes are identified. Then, the differences in genomic features between the cell types are observed to justify the differences in expression.

The process starts by constructing a dataset (Fig. 2a and 3a). TFs, that fall within the promoter region of genes, are first mapped to their corresponding binding sites on the genome. Then, they are associated to the genes and become the instances, while the genes themselves become the bags. The instances are described with features calculated from the biological data. Up to this step, the process is the same for the two scenarios. For the second scenario, appropriate horizontal concatenation is used for the features of each cell type (Fig. 3a).

Afterwards, gene expressions are discretized to be used as the training labels during classification. For the first scenario, that is enough (Fig. 2b). For the second scenario, there is an extra step as the labels are combined according to a pattern of expression (Fig.

3b). For example, if a gene is expressed in one cell type but not in another, the pattern "1-0", would be combined to a label 1, 0 otherwise.

MIL classification is performed and the output are the predicted gene expression labels. The secondary output are the concepts, i.e. the TFs responsible for gene expression (Fig. 2c and 3c), for the single cell type scenario and the TFs responsible for differential regulation, for the multiple cell type scenario. By examining those, the genomic properties that made them significant can be determined.

In Section 2, the data used to construct the dataset along with the gene expression discretization strategies and the MILES (Chen *et al.*, 2006) classifier are discussed. In Section 3, the choice of a subset of genes is explained and the experiments performed are presented in detail along with the results. Finally, in Section 4, discussion points are given for ways that could improve the method and some future prospects.

2 MATERIALS AND METHODS

In this section, the first steps of the method will be discussed, namely the available data (Subsection 2.1) and the construction of the dataset (Subsections 2.1.1 and 2.1.2) and the gene expression discretization strategies (Subsection 2.1.3). Furthermore, as the prediction of the instance labels is considered the most interesting for this study, the MILES classifier, Multiple Instance Learning via Embedded Instance Selection (Chen *et al.*, 2006), was viewed as the best candidate for the task, as it can highlight multiple concepts instead of just one. This will be discussed in Subsection 2.2.

2.1 The data of the ENCODE Project

The Encyclopedia of DNA Elements (ENCODE) Consortium (Consortium, 2004) is a collective effort between researchers to create a database containing all the information that describe the genomic "landscape". This is done by identifying and measuring all the functional and regulatory elements that control gene expression and determine cell fate.

The data in ENCODE is organized in Tiers of cell types, according to the priority of each cell type for new experiments. All cell types, for which experiments have been conducted, are organized in 3 Tiers, with Tier 1 having top priority. Tier 1 contains 3 cell types; a healthy blood cell (GM12878), a cancerous blood cell (K562) and a stem cell (H1-hESC, H1 human Endothelial Stem Cell). Since Tier 1 has top priority, most of the experiments have already been performed for these 3 cell types. For the purpose of this study, data for the 3 cell types of Tier 1 was used.

The first information that was needed, was the TF ChIP-seq uniform peak data, that identify the regions of the genome where certain proteins were observed to have bound. The gene expression data (RNA-seq from ENCODE/Caltech) were obtained for all 3 cell types of Tier 1 and, also NHEK, a skin cell. Then, to construct the features of the dataset, data for DNA methylation (DNA Methylation by Reduced Representation Bisulfite Seq from ENCODE/HudsonAlpha), Open Chromatin (DNase-seq peaks) and Histone Modifications (Uniform Histone peaks) were collected (Links to all data used are provided in the Supplementary Material, Section S-1).

2.1.1 Construction of the dataset To associate TFBSs to genes, a TFBS was called for each binding site in the 1 Kb upstream region of a gene (Fig. 4). For each TFBS ChIP-seq peak, if the start fell within the range of the thousand bps of a promoter region, then the TFBS was associated to that gene. With this criterion, 660,000 TFBSs were associated to genes.

Two filtering steps were needed, one before associating TFBS peaks to genes, and one after. As a first step, filtering of the peak data was performed for actual TFs, as not all the binding proteins, for which ChIP-seq experiments were conducted, are TFs. Based on a study of human TFs (Vaquerizas *et al.*, 2009), out of 146 unique experiments, for the 3 cell types of Tier 1, 82 were recognized as TFs. For these, the complete concatenation of all experiments (4.1 M lines) was filtered.

As a second filtering step, the resulting data was filtered for duplicates, due to the way ChIP-seq works. Experiments were performed for a range of conditions, for certain TFs. Also, for the same TFs and conditions, results may be available from several institutes. So, for a certain TFBS associated to a certain gene, multiple locations may be available that differ very little (less than 20 bp). To account for those, only one was selected based on the highest q-value. This resulted in the final dataset of approximately 230,000 instances (TFBSs) associated to 15,008 bags (genes).

2.1.2 Features

Binary Features The information that ChIP-seq experiments provide is that a particular TF was observed to be bound on the genomic location where the peak is observed. A set of 82 binary features was made to represent which TF the TFBS was associated with. A binary feature that signified whether a TFBS was ever encountered in a particular cell type was, also, used while constructing the dataset.

Genomic Distances Two features were calculated. The distance between a TFBS was measured in two ways; one in terms of proximity to the TSS relative to the other TFBSs associated to that gene and one counting the absolute distance in bps, as can be seen in Fig. 5.

DNA Methylation For this data, which consist of single methylated bps, 3 features were calculated for each TFBS; the sum of methylated bps within the TFBS and in windows of 50 and 100 bps on either side of the TFBS, as can be seen in Fig. 6.

Open Chromatin (OC) For this data, obtained with DNase-seq, the peaks represent regions where the DNA sequence can be cut by the enzyme DNase I, implying that this region is open and accessible by other molecules. These peaks span long DNA regions, so the two features calculated for each TFBS were the number of OC peaks under which a TFBS might fall and the distance, in bps, from the center of the peak. If multiple peaks were assigned to one TFBS, then the maximum distance would be assigned.

Histone Modifications (HM) The same strategy was employed for this data as for OC. Three datasets were explored because of their association to gene activity (Fig. 1c and (Kooistra and Helin, 2012)), yielding 6 features per unique cell type.

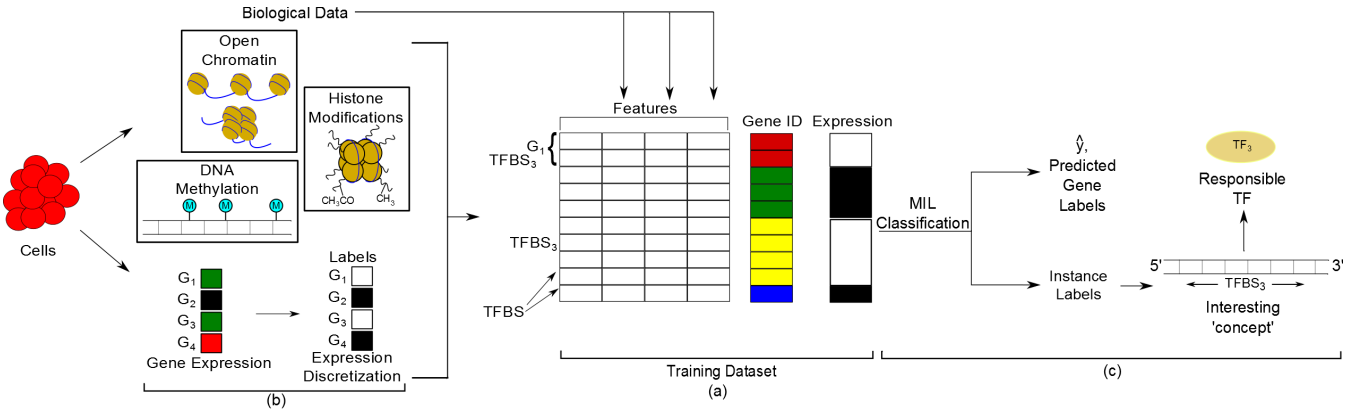


Fig. 2: **Single cell type scenario**: Methodology used for TF discovery for a group of – or all – genes in a single cell type. **a**: Construction of the dataset from the biological data. **b**: Discretization of gene expression to use as labels. **c**: MIL classification and identification of interesting TFs for gene expression.

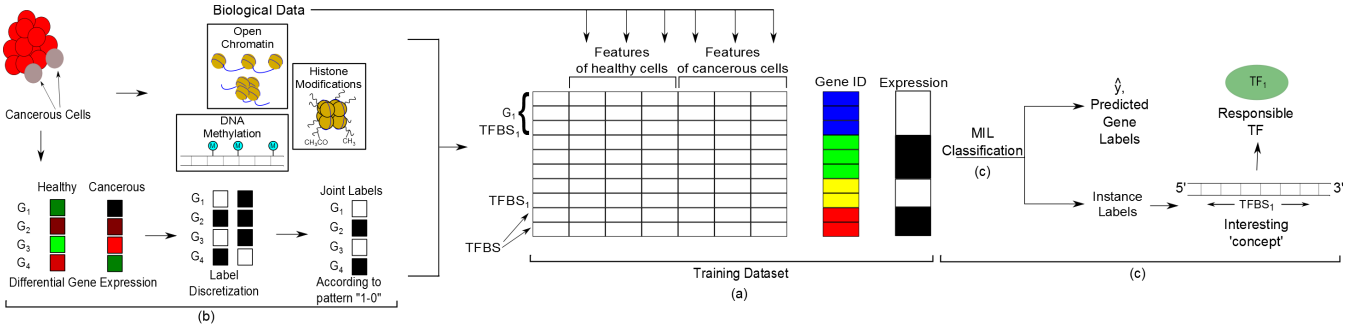


Fig. 3: **Multiple cell type scenario**: Methodology employed for TF discovery for a group of – or all – genes in two cell types, one healthy and one cancerous. The expression pattern "1-0", i.e. expressed in healthy, but not expressed in cancerous cells, is of interest in this example. **a**: Construction of the dataset from the biological data. The feature vectors, describing the TFBSs, are concatenated appropriately for the number of unique cell types. **b**: Discretization of the expression levels with the extra step of label combination. **c**: MIL classification and identification of TFs responsible for differential regulation.

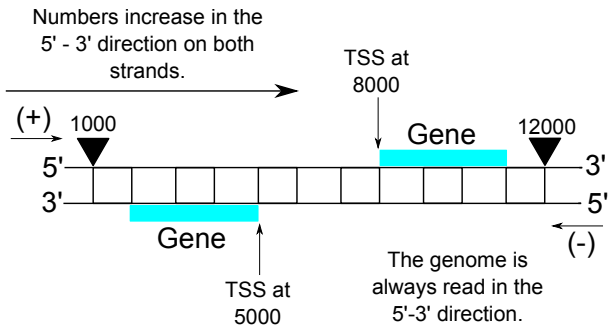


Fig. 4: The TSSs for two genes on the positive and negative strands of a short, example DNA sequence. The numbering of bps increases on the 5'–3' direction on both strands.

ChIP-seq Features Finally, four features of the original ChIP-seq peak data were used. These were the peak intensity normalized by ENCODE, the assigned p-value and q-value of the peak and the peak centre offset.

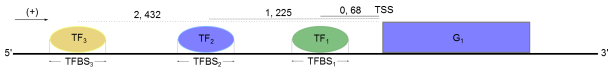


Fig. 5: Calculation of genomic distances. Each TF is assigned an integer, starting with 0, according to its proximity to the TSS. The absolute distance from the TSS, in bps, is also measured.

Fig. 6: Calculation of the DNA methylation features for each TFBS.

In total, 100 features described each TFBS, with 12 of these being unique for a certain cell type. Table 1 gives an overview of the features generated.

2.1.3 Label Discretization Gene expression is measured in real, positive numbers, called RPKM (Reads per Kilo base per Million mapped reads). RPKM is proportional to the abundance of each

Number of Features	Type	Description
4	Float	TFBS ChIP-seq peak data
82	Binary	TF IDs
1	Binary	Presence of each TF
2	Int	Genomic Distances
3	Int	Methylation Counts
2	Int & Float	Counts of and distances from associated OC peaks
6	Int & Float	Counts of and distances from associated Histone Modification peaks
Total: 100		

Table 1. The features generated from the ENCODE data per cell type.

gene or transcript. It is a normalized value that corrects for the library size and reference sequence length.

The RPKM values may not be directly comparable between cell types, as a particular cell may have a higher protein production in general. For this reason, a pre-processing normalization step was applied first to equalize the expression values between all cells. For every cell type, the RPKMs of all genes were added together. This sum represents the overall RNA production of a cell type. Each value was normalized by the mean of those sums:

$$N_c(g) = R_c(g)\frac{C}{\sum_{k\in G} R_c(k)} \tag{1a}$$

$$\text{where, } C = \frac{1}{|E|}\sum_{e\in E}\sum_{k\in G} R_e(k) \tag{1b}$$

where $e \in E$ are the different cell types, with c a cell type of interest. g is the gene in consideration, over all possible genes $k \in G$.

This new value, $N_c(g)$, for a gene in a particular cell type, resembles the TPM (for Transcripts Per Million) values, that can be generated from RPKM, by dividing with the sum of RPKMs for all genes and multiplying by a million, that is stated to be a more accurate measure of RNA total production (Wagner *et al.*, 2012).

The problem with RPKM values is that they are unfit to be used as labels, since they are continuous. So, two discretization strategies were tried to discretize these values into binary labels to represent expression and to use them for classification.

For both strategies a threshold is used. To justify the selection of a threshold, it must be emphasized that the focus of this study is not on the performance of classifiers, but on the extraction of meaningful information that can be linked to biology. To achieve this, it will be judged if the MIL framework is able to answer such biological questions. For this reason, the method used here was to measure the performance, while varying the thresholds, and select the one for which classification worked best.

$\mathbf{y}_\theta$ ***Discretization*** The first strategy to make binary labels out of RPKM was to set a threshold, θ, to the real values, below which gene expression would be assigned a *0* label and a *1*, otherwise (Eq. 2). The main assumption in this part is that at some point during the cell cycle, a gene will be expressed. This was used as a rule of thumb to keep in mind while testing the different thresholds. The threshold θ was varied between values 0 and 80, in RPKM $(0, 7.5, 12.5, 17.5, 20, 30, 40, 50, 60, 70, 80)$.

$$y_\theta = \begin{cases} 0 & \text{if } E_{ic} \leq \theta \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

where E_{ic} is the expression level of gene i for a particular cell type c.

$\mathbf{y_{010}}$ ***Discretization*** The second strategy was to use a comparison between the expression levels in the 3 cell types of Tier 1 and another cell type used as reference, in this case NHEK, a human skin cell type. The comparison scheme tested signified that a gene's expression is not significantly different from the one in the reference (scheme "010", Eq. 3).

$$y_{010} = \begin{cases} 0 & \text{if } E_{ir} = 0 \wedge E_{ic} = 0 \\ 1 & \text{if } (1-t)E_{ir} \leq E_{ic} \leq (1+t)E_{ir} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where E_{ir} is the expression value of gene i in reference cell type r, E_{ic} is the one in cell type c that is being compared and t is the similarity threshold as a percentage. The second strategy has the advantage that the labels are intuitively understood. More genes having positive labels signifies similar behaviours, or expression patterns, to the reference skin cell. The threshold t was varied between a percentage of 0.3 and 0.9 in increments of 0.1.

To investigate the range of the thresholds, a distribution of the expression levels of all genes of all 4 cell types (3 plus reference skin cell) was built (Fig. 7). It can be seen that most of the genes are very lowly expressed.

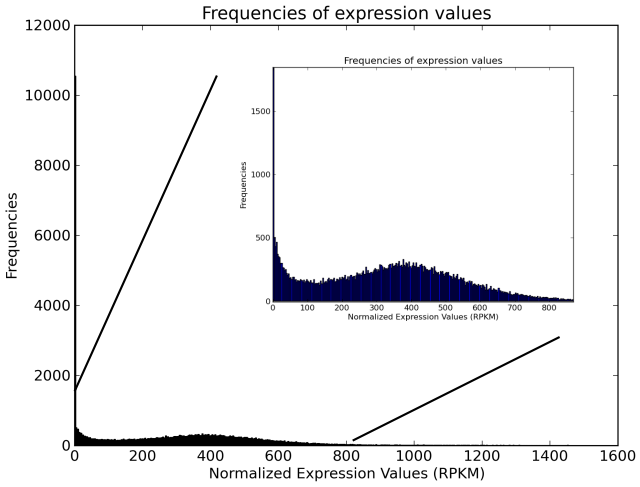
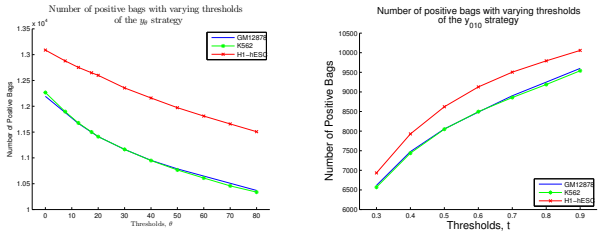


Fig. 7: Distribution of normalized expression values, in RPKM, for all 4 cell types. Most of the genes are lowly expressed. The distribution is zoomed into, in the inlay picture.

For the y_θ strategy, as θ increases, it becomes more strict and more bags are labelled negative (Fig. 8a). For the y_{010} strategy, it is the opposite (Fig. 8b). For patterns of expression, such as

"a gene is expressed in one cell type but not in another", this can work in reverse if the pattern contains a zero. For example, for the y_θ strategy, as more genes are labelled negative, they have more chances of fitting a particular pattern.



(a) (b)

Fig. 8: **Number of positive bags to thresholds**: **a**: Inverse proportionality to θ. **b**: Proportionality to t.

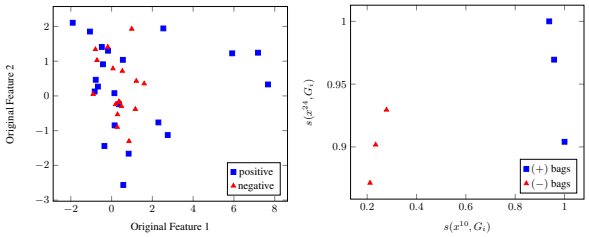
2.2 MIL and the MILES Classifier

The advantage of the MIL framework is that it can identify a "concept", an instance that is the most characteristic, or descriptive, of the positive class. The advantage of MILES (Chen *et al.*, 2006) over other classifiers, is that it can perform instance selection and not select just one, but multiple concepts. This functionality is beneficial for this study, as multiple TFBSs can be picked out as significant, thus reflecting the biological truth well, as more than one TFs, working in unison, may be responsible for gene expression.

What MILES does is firstly mapping all the bags into a similarity space, by measuring the minimum distance between an instance, x^k, and all other instances, x_{ij}, in all bags, G_i. This similarity between a bag, G_i, and an instance x^k is defined as:

$$s(x^k, G_i) = \max_j \exp\left(-\frac{||x_{ij} - x^k||^2}{\sigma^2}\right) \tag{4}$$

This creates a space with dimensionality equal to the number of instances. In this space, every bag is represented by a single point, making the classification problem easier (Fig. 9). That point has coordinates m, where m is a vector of length equal to the number of instances. Each coordinate is one of the similarities $s(x^k, G_i)$. Then MILES tries to find a linear classifier $y = sign(w^T m + b)$, where y are the class labels, w is a vector of weights and b the distance of the resulting hyperplane from each class. To find a hyperplane that best separates the two classes, MILES utilizes an L1-SVM. Any component, w_k, of vector w that is non-zero, indicates the significance of the effect of the k^{th} instance on the classifier. Therefore, the instances whose weights are non-zero are more helpful to classification. Fig. 9 illustrates an example of this process. In Fig. 9a, the 44 instances of 3 positive and 3 negatives bags can be seen in the original feature space. After calculating the similarities of each instance, x^k, to every bag, G_i, the L1-SVM returns non-zero weights for $s(x^{10}, G_i)$ and $s(x^{24}, G_i)$, marking instances 10 and 24 as significant. In the 2D similarity space of these 2 instances (Fig. 9b), the bags are depicted as individual points and their classification becomes trivial.



(a) (b)

Fig. 9: **Mapping of bags in instance space: a**: A toy example of 6 bags, 3 of which are positive and 3 negative, with 9,8,7,5,9 and 6 instances respectively (44 in total). Some instances are on top of each other. **b**: $s(x^{10}, G_i)$ and $s(x^{24}, G_i)$ had non-zero weights and were selected from the SVM out of the 44. The 6 bags are mapped onto this 2D space. Classification becomes trivial.

Usually, SVMs use the squared 2-norm of the weight vector $||w||$, but this results in a Quadratic Program to solve. MILES uses the 1-norm SVM, $||w||_1 = \sum_k |w_k|$, which is easier to solve, as it is a Linear Program. Slack variables, ξ, η, are used to account for overlap between positive and negative bags and the error of these false negatives (FN) and false positives (FP) must be minimized. Using different penalties between these slack variables, C_1, C_2 respectively, can correct for class imbalances, which is the parameter $C = {}^{C_1}/_{C_2}$ in the implementation of MILES in the MIL Toolbox (Tax, 2013). The 1-norm SVM is formulated as:

$$\begin{aligned} \min_{w,b,\xi,\eta} \lambda \sum_{k=1}^{n} |w_k| + C_1 \sum_{i=1}^{l^+} \xi_i + C_2 \sum_{j=1}^{l^-} \eta_j \\ \text{s.t.} (w^T m_i^+ + b) + \xi_i \geq +1, i = 1, \ldots, l^+, \\ -(w^T m_j^- + b) + \eta_j \geq +1, j = 1, \ldots, l^-, \\ \xi_i, \eta_j \geq 0, i = 1, \ldots, l^+, j = 1, \ldots, l^- \end{aligned} \tag{5}$$

where, C_1 and C_2 are the penalties for FN and FP, respectively, l^+, l^- are the numbers of positive and negative bags and i, j are the indices of the positive and negative bags.

3 RESULTS

3.1 The WNT Pathway

Searching for differences in genomic features throughout the complete dataset may be meaningful between two cell types for a specific pattern of expression, but not so for a single cell type. For the latter, it is more meaningful to search for significant TFs in a subset of genes related to a specific function. For this reason, a subset of genes was needed that should have been studied well enough in order to be able to relate the findings of this study, if any, back to biology.

The signalling pathway chosen according to these criteria was the WNT pathway. This pathway is one carrying chemical signals between cells. As such, it is observed only in multicellular organisms and is not found in single cell ones, like bacteria. It is

highly conserved, meaning that it is very similar in very different organisms. The chemical signal conveyed between cells is that of $\beta-$catenin, a protein that regulates gene transcription inside the cell nucleus and coordinates cell-cell adhesion. The up- or down-regulation of the pathway, or its disruption, has been implicated in various diseases including cancer. It has been studied intensively for the past 30 years and, as will be seen in Subsection 3.4, there was some information to link the present findings to the underlying biology.

From the original dataset of 15K bags (genes) with 230K associated instances (TFBSs), 125 bags, with 2004 instances, are associated to GO term GO:0016055, that is the WNT pathway. This small dataset was used in all of the following experiments as a validation platform.

3.2 Experiments for Label Discretization

For all 3 cell types of Tier 1, the labelling thresholds were varied – for both strategies – and the performance of 3 classifiers was evaluated. The pattern "G0 K1" was also investigated. This pattern signifies different things for the two strategies. For the y_θ strategy it means that a gene is not expressed in a healthy blood cell (GM12878), but it is expressed in a cancerous one (K562). For the y_{010} strategy it signifies that a gene is different from the one in the reference for the healthy blood cell, while it is not so different for the cancerous blood cell.

The 3 classifiers used were: (*a*) Citation MIL, (*b*) Simple MIL with a log linear classifier and. (*c*) MILES with a radial kernel of optimized distance and FN-FP penalty threshold parameter $C = 0.005$. Citation MIL is a MIL version of nearest neighbour, where the classifier uses the Haussdorff distance between the bags of instances. The final bag label is given by majority voting of the labels of the K nearest bags. Simple MIL is trained on the complete dataset, disregarding the fact that it is organized in bags. When evaluating, all the instances of a bag are classified and, by combining the labels of the highest 1%, a label is given to the bag. For Citation MIL and MILES scaling of the features to unit variance was applied, apart from the binary ones.

For cell type GM12878, the highest performance is achieved by MILES over the other two classifiers over all experiments, as can be seen on Fig. 10a. The performance is stable with small standard deviation over the 5 folds for each cross validation experiment. Overall, the first strategy of the θ thresholds outperforms the second, y_{010}, one apart from the second classification scenario of pattern discovery (pattern "G0 K1"). An immediate difference between the two strategies is the number of positive bags (Fig. 8). For the y_{010} strategy, even at 90% similarity, i.e. a gene is given label 1 if its expression is within 10%-190% of the reference, roughly 9500 genes out of 15000, per cell type, are labelled positive (Fig. 8b), which immediately implies that the cell types are not similar enough for comparisons.

Considering each cell type (Figs. 10 and 11), the classification performance varies, because of the differences in genomic features. Different thresholds work best for each cell type; lower thresholds θ for the first strategy, allowing more genes to be positive, yield better performance for the blood cells, while, for the stem cell – where most of the genes are highly expressed (Fig. 8) – higher thresholds counteract the positive class imbalance and yield better performance.

For the second strategy, higher thresholds t, allowing more genes to be labelled positive, help boost performance as they counteract the negative class imbalance. The y_{010} strategy is, overall, more robust both performance- and threshold-wise, as for all cell types and the pattern a value of $t = 0.7$ works reasonably well (almost 75% AUC); even for H1-hESC, where the best performing threshold was $t = 0.3$, the next best one was $t = 0.7$.

A summary of the best performance for MILES and the corresponding thresholds, θ, t, is given in Table 2.

	Discretization Strategies			
	y_θ Strategy		y_{010} Strategy	
Cell Type	θ	AUC	t	AUC
GM12878	12.5	90.0% (10.5%)	0.9	78.9% (9.0%)
K562	12.5	82.6% (8.8%)	0.8	73.5% (9.8%)
H1-hESC	70	85.0% (9.7%)	0.3	75.9% (9.4%)
G0 K1	20	72.6% (20.1%)	0.9	83.6% (20.8%)

Table 2. Highest MILES performance for each cell type and corresponding thresholds.

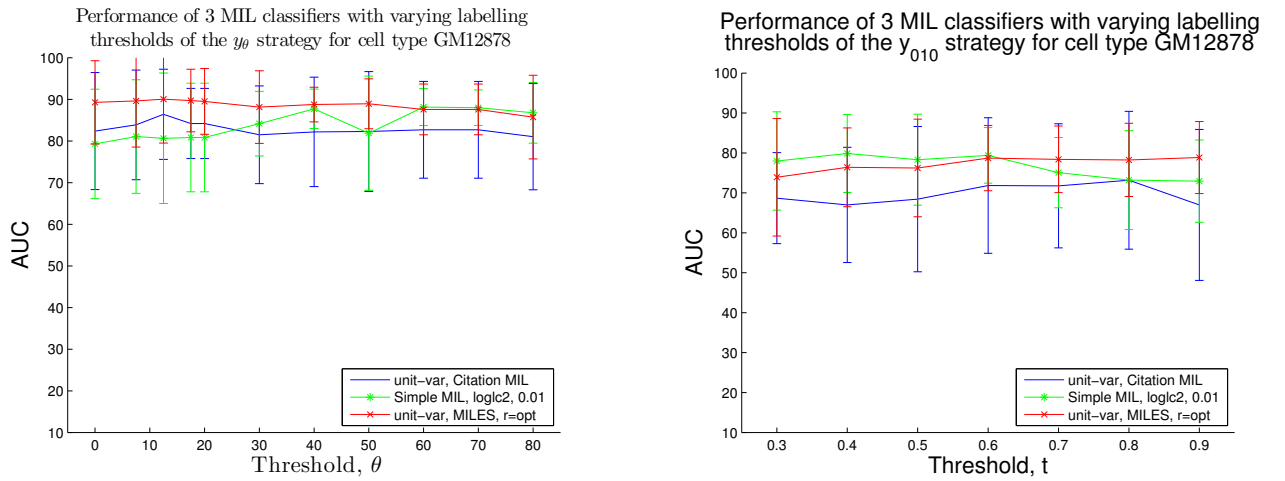
3.3 Experiments for Parameter Optimization

An experiment was conducted for two datasets of the healthy blood cell, GM12878, for each of the labelling strategies. 5-fold cross-validation was performed on these datasets varying the penalty between false positives (FP) and false negatives (FN), which is a parameter of MILES. C was varied in an exponential manner $(0.0001, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, 100)$. The classification performance is given by the AUC (Area under ROC curve). For the varying of C, the following figures are plotted: (*a*) AUC against C (Figs. 12a and 12b), (*b*) mean error against C (Fig. 12c and 12d). The distance of the radial kernel is optimized internally by MILES. Here, it is defined as the first nearest neighbour distance, of the complete training set.

As can be seen in Figs. 12a and 12b, the performance increases after the initial values, for both strategies, and stabilizes after $C = 0.01$ (also, Fig. SF-4 in Supplementary). A similar picture (Figs. 12c and 12d) is drawn for the mean classification error, as it decreases and stabilizes after $C = 0.01$.

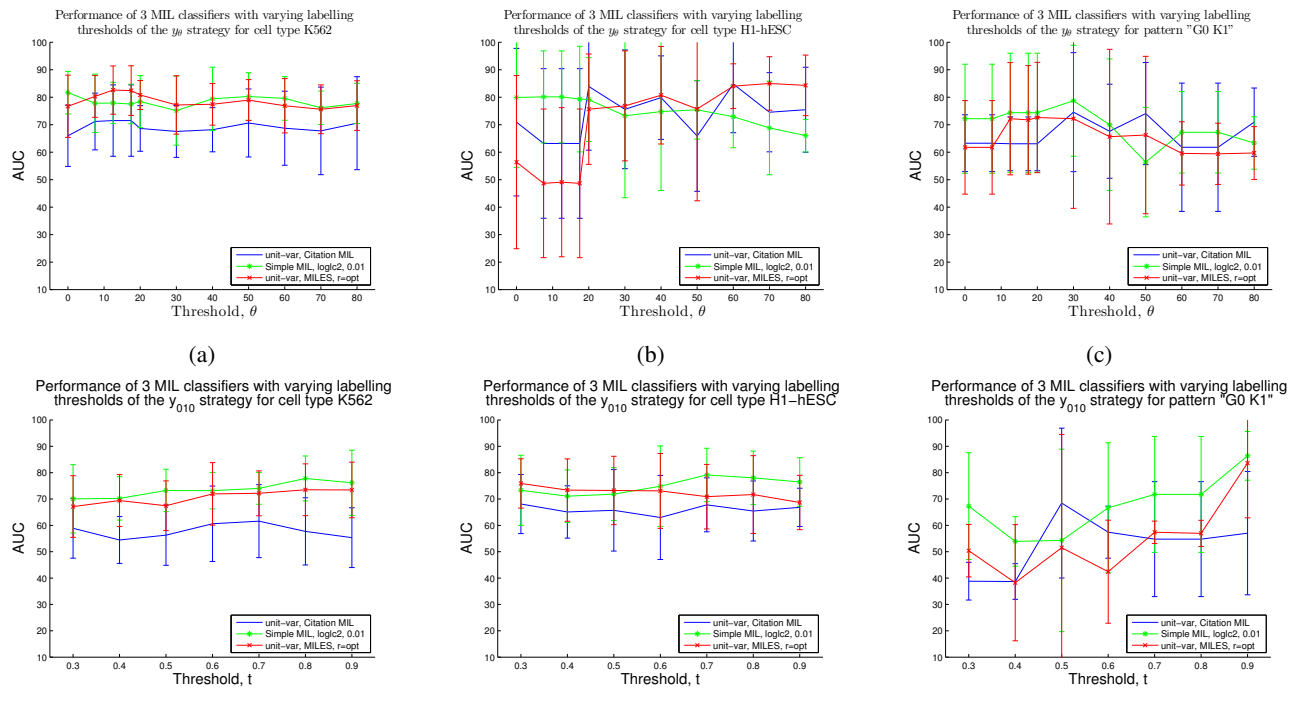
3.4 Result Interpretation

The normalization step did not make much difference (Figs. 10, 11 and SF-2, SF-3 in the Supplementary; also Tables 2 and ST-1 in the Supplementary). In some cases it decreased performance very slightly. It marginally improved the performance for cell types K562 and H1-hESC, that were the ones with higher total RNA production and only for y_{010}, the second labelling strategy. For these two cell types, the normalization step decreased the gene expression levels, providing a fairer comparison between them and the reference skin cell, thus labelling the bags more correctly. For the first strategy, y_θ, it shifted the thresholds of best performances, which is logical, since it shifted the values for every cell type according to total over- or under-production of RNA. Applying threshold, θ, afterwards



(a) (b)

Fig. 10: **Classifier Performance (GM12878)**: **a**: For the y_θ strategy. **b**: For the y_{010} strategy.



(a) (b) (c)

(d) (e) (f)

Fig. 11: **Classifier Performance for cell types K562, H1-hESC and pattern "G0 K1"**): **a**, **b**, **c**: For the y_θ strategy. **d**, **e**, **f** : For the y_{010} strategy.

resulted in more genes labelled with 0 or 1 according to over- or under-production, respectively.

The only reversal in performance, is for pattern "G0 K1", where the y_{010} strategy outperforms the y_θ one, for the most relaxed threshold $t = 0.9$. For the lower thresholds, t, the y_{010} strategy performs worse than random (Fig. 11f).

Fig. 13 depicts: (a) the number of instances (TFBSs) selected by MILES against C, for both strategies (Fig. 13a and 13b), (b) the corresponding number of unique TFs selected by MILES, out of 79, against C (Fig. 13c and 13d). A cut-off for the weight absolute value of 10^{-6} was used to only focus on the most significant instances. Two scatter plots are, also, given of the AUC against the

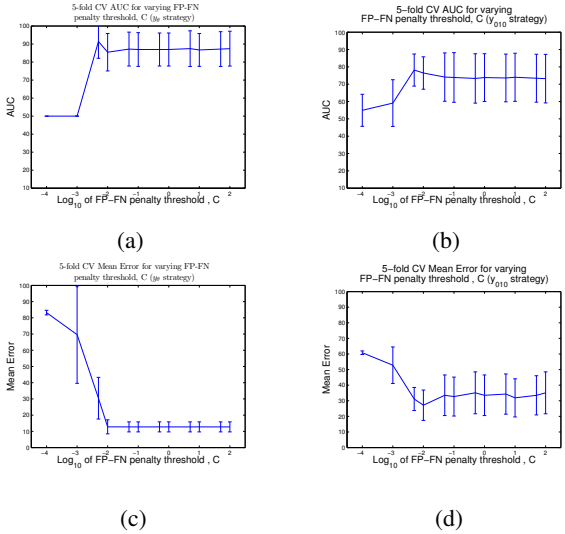


Fig. 12: **Effects of varying C on performance and mean error of MILES**: Mean 5-fold AUC for: **a** the y_θ labelling strategy at threshold 7.5 and for **b** the y_{010} strategy at threshold 0.8. **c**, **d**: The respective mean errors. The errorbars are the standard deviations of the 5-fold cross validation.

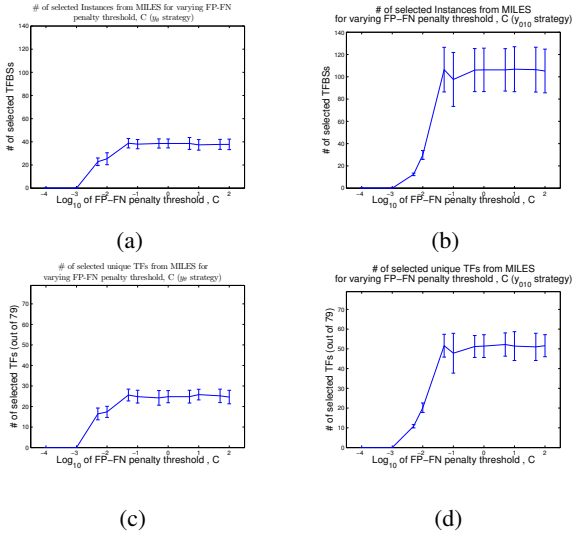


Fig. 13: **Correlation of selected instances to FP-FN penalty threshold,** C: Number of selected instances (TFBSs) by MILES (weight cut-off, $|w_k| > 10^{-6}$) over 5 folds for various thresholds of C for **a**: the y_θ labelling strategy and for **b**: the y_{010} one. **c**, **d**: Corresponding numbers of selected TFs for both strategies.

number of selected instances (Figs. 14a, 14b) and corresponding TFs (Figs. 14c, 14d).

The varying of C seems to have an effect on the number of instances that are selected as significant from MILES. With lower thresholds, MILES is not able to find an optimal solution and the number of TFs selected as significant is zero or very small. A note on the standard deviations for Fig. 13 is that the number of selected features varied greatly between the different folds for every value of C. That was why a cut-off value of 10^{-6} was used for the weights w_k and the picture became much clearer. Between the two discretization strategies, y_θ works better, as less TFs are selected on average (Figs. 13c and 13d) out of the 79 possible.

The best performing threshold was $C = 0.005$ and was chosen over $C = 0.01$, although the error is less (Fig. 12c), and the performance more stable (Fig. 12a). The number of instances selected by MILES as important is smaller for $C = 0.005$ over 5 folds than $C = 0.01$ (Fig. 13a).

Between the two strategies (Figs. 14a and 14b), fewer selected instances lead to improved performance. This is the combination of conclusions from Figs. 12a and 13a, for the y_θ strategy and Figs. 12b and 13b, for the y_{010} strategy, but it does not mean that there is necessarily a correlation between the selected number of instances by MILES and performance. While varying C, the performance is stable, apart from the cases where MILES cannot find any concepts ($C = 10^{-4}$ and $C = 10^{-3}$).

3.4.1 Single Cell Type Scenario For the single cell type classification scenario, the classifier was retrained for the complete dataset, without fold separation. MILES returned 792 instances as important for gene expression, out of 2004. These instances belonged to 73 unique TFs and 105 genes. Of these instances,

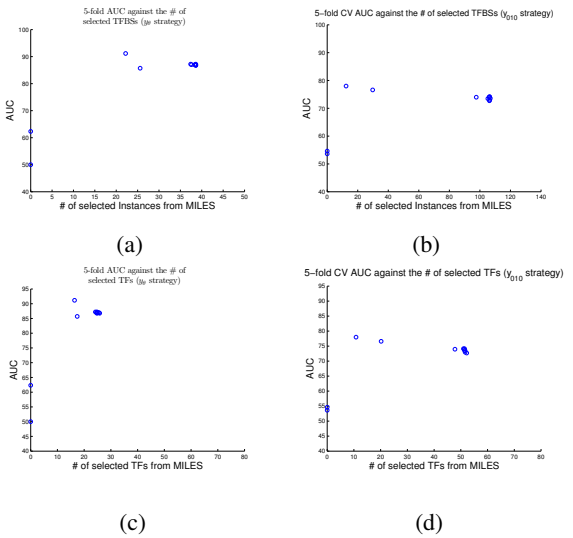


Fig. 14: **MILES performance (AUC) to number of selected TFs**: **a**, **b**: Mean AUC against mean number of selected instances by MILES over 5 folds for the y_θ and the y_{010} labelling strategies. **c**, **d**: Corressponding number of selected TFs.

only 21 had weights $|w_k| > 10^{-6}$, out of which 12 were positive weights. These 21 highly weighted TFBSs belong to 19 unique genes. As will be seen in Subsection 3.4.2, some of these genes (HIC1, MITF, AMOTL1) are also important in the two cell type scenario, suggesting their general importance for the WNT pathway.

MILES performed well and was very flexible. Flexibility here means that instances were picked as important even when the TFs were not bound on the promoter region of a certain gene, reflecting the biological reality of the negative effect some TFs may have

on gene expression; when a TF is not bound, it allows a gene to be expressed. Also, instances with negative weights were picked, meaning that, though they were bound, they were influencing gene expression also in a negative manner.

3.4.2 Two Cell Type Scenario For the two cell type pattern classification scenario, the results for pattern "G0 K1" for the second strategy, y_{010}, at threshold $t = 0.9$ will be discussed here in further detail. The classifier was retrained for the complete dataset, without fold separation. As a result, 138 instances were selected as important, belonging to 53 unique TFs and 24 unique genes. Out of these 138 TFBSs, only 10 had a weight absolute value over 10^{-6}, i.e. $|w_k| > 10^{-6}$. For these 10 TFBSs that belong to 8 TFs and 9 genes, the differences in genomic features can be seen for the 2 cell types in Table 3. The 24 unique genes are known genes related to the (de-)regulation of the WNT pathway and have involvment in cancer whether as activators or repressors.

As can be seen in Table 3, the most meaningful features for differential regulation turn out to be whether a TF has been encountered in one cell type or not, the open chromatin interactions and the histone modifications. These are given in bold, to signify that they were different between cell types. There is evidence in the literature that support the important role of these features both for cancer and the WNT pathway (Kandoth *et al.*, 2013), (Liu *et al.*, 2008). The genomic distances can also be considered as an interesting result. It is believed that the first TF after the TSS is the most responsible for gene expression. Here others are found that are further away.

For the 24 genes, whose instances were found important for gene expression by MILES, a gene enrichment analysis for TFs was performed with the web-based application ChEA, ChIP Enrichment Analysis (Lachmann *et al.*, 2010). This application uses a database of 190K ChIP interactions describing the binding of 92 TFs to almost 32K genes. With this database, one can find TFs most likely responsible for gene expression changes by computing the over-representation of their TFBSs.

After searching in ChEA for the 24 genes whose instances were found important for gene expression by MILES, the TFs that came out with corrected p-values lower than 0.05, could not be rejected as random for gene expression changes. These can be seen in Table 4 with the corresponding weights, w_k, assigned by MILES.

The second highest weighted instance by MILES, for TF EZH2 (enhancer of zeste homolog 2), turns out to be likely significant for the expression of gene HIC1 (ENSG00000177374) in mice. When the same search was performed for humans only, EZH2 did not appear in the results, meaning it is a novel finding of the current research. Evidence of interactions between EZH2, SUZ12 and HIC1 could be found in the literature (Boulay *et al.*, 2012), although the interaction is not of the same kind, meaning TFs bound on the promoter region of a gene. In the case of Boulay *et al.* (2012), EZH2, SUZ12 and HIC1 form a complex to recruit polycomb proteins. Their experiments were performed on mice, but their findings are transferable to humans, since this gene is conserved between humans and mice [1]. Further evidence could be found in the literature that when EZH2 binds in the promoter region of HIC1, it causes di- or tri-methylation of H3K27, which silences the gene (Svedlund *et al.*, 2012). This explains both the observed features of that TFBS but also justifies the current findings. This is a very encouraging result as it shows that the current findings have some merit in biology.

It is, also, interesting that this result was still obtained, while the labels do not quite reflect the "truth" of gene expression. The expression values of gene HIC1 were 562, 51, 0 and 67, in RPKM, for GM12878, K562, H1-hESC and NHEK, respectively. Therefore, with the y_{010} labelling strategy, gene HIC1 was labelled 0 for GM12878 as it was significantly different from the reference ($0.1 * 67 < 562 \not< 1.9 * 67$), while it was labelled 1 for K562, as it was similar to NHEK ($0.1 * 67 < 51 < 1.9 * 67$), therefore falling into pattern "G0 K1" and labelled 1.

Other TFs were likely significant as well for humans in the enrichment analysis, but had been assigned low weights by MILES. On the other hand, some instances with higher MILES weights, w_k, are not significant in the enrichment analysis ($p > 0.05$), but they do appear nonetheless.

4 DISCUSSION

A method was presented to investigate the genomic differences between different cell types that influence gene expression, using the MIL framework. To the authors' knowledge, this is a novel approach. In general, MIL has started been applied to biological problems only very recently (Li *et al.*, 2013), (Eksi *et al.*, 2013).

From a performance point of view, the results are promising. The MIL framework seems an adequate platform to address such biological questions. The MILES classifier, which was deemed as the most adequate to address the question of this study, performs well in all different scenarios. From a biological point of view, a partial confirmation could be obtained for some instances; they turn out to be likely significant for gene expression in humans and mice, just as MILES predicted. This gives motivation for further investigation.

Some remarks have to be made regarding the data that can lead to improvements to make the method more robust.

For some experiments in the ENCODE database, an experiment for a TF may have been done only for one cell type. In this case, if a TF is found to bind in this one cell type, nothing can be said for the others, because the information does not exist. On the other hand, if the experiment was done for all 3 cell types of Tier 1, the information is there to store in a binary feature. The problem with such a feature is that a zero has an ambiguous meaning, as it is not known if the TF was not found to be there or because the information does not exist. Therefore, this feature must signify whether a TFBS was ever encountered in a particular cell type, because this definition accounts for this ambiguity. If a TFBS was unique, zeros were assigned to the cell types for which information did not exist. Another way to address this ambiguity could be to remove the experiments that were not performed for all 3 cell types. This is true for 27 out of the 146 unique ChIP-seq TF experiments, though a lot of data would be dismissed.

Another important step, would be to perform the classification experiments again with the inclusion of the peaks that belonged to binding proteins that were not TFs according to Vaquerizas *et al.* (2009). Some confusion seems to exist on the adequate definition of a TF, as some proteins that were excluded from the current study,

[1] http://www.ncbi.nlm.nih.gov/homologene/4740

TFBS	TFBS Characteristics		Cell Type	Genomic Features						Gene ID	Label
	Weight	Position after TSS		Observed with Gene	Methylation	Open Chromatin	Histone Modifications H3K4	H3K27	H3K36		
E2F6	2.029	14	GM12878	**No**	5% (3%)	2	1	0	0	ENSG00000015133	1
			K562	**Yes**	5% (3%)	2	1	0	0		
EZH2	2.003	1	GM12878	No	6% (0%)	**3**	1	0	0	ENSG00000177374	1
			K562	No	6% (0%)	**2**	1	0	0		
CTCF	1.883	1	GM12878	Yes	18% (3%)	None	0	**1**	0	ENSG00000133067	1
			K562	Yes	18% (3%)	None	0	**0**	0		
ZBTB7	1.71	2	GM12878	**No**	8% (4%)	1	**0**	**1**	0	ENSG00000143816	1
			K562	**Yes**	8% (4%)	1	**1**	**0**	0		
ZNF143	1.397	3	GM12878	**No**	None	**0**	**0**	**1**	0	ENSG00000153071	1
			K562	**Yes**	None	**1**	**1**	**0**	0		
E2F6	1.269	2	GM12878	**No**	None	**0**	**0**	**1**	0	ENSG00000165795	1
			K562	**Yes**	None	**1**	**1**	**0**	0		
ELF1	1.246	7	GM12878	**No**	6% (2%)	**0**	**0**	**1**	0	ENSG00000166025	1
			K562	**Yes**	6% (2%)	**1**	**1**	**0**	0		
ZNF143	1.094	9	GM12878	Yes	8%, (2%)	1	1	**1**	0	ENSG00000187098	1
			K562	Yes	8%, (2%)	1	1	**0**	0		
STAT1	0.691	15	GM12878	**Yes**	None	None	1	**1**	0		
			K562	**No**	None	None	1	**0**	0		
RUNX3	-0.016	13	GM12878	**Yes**	3% (0%)	**2**	1	0	0	ENSG00000119509	0
			K562	**No**	3% (0%)	**1**	1	0	0		
EZH2	1.93	1	GM12878	**No**	12% (8%)	**3**	1	**0**	0	ENSG00000177374	1
			K562	**No**	12% (8%)	**2**	1	**0**	0		
			H1-hESC	**Yes**	12% (8%)	**3**	1	**1**	0		

Table 3. Summary of the differences in genomic features for the highest weighted instances by MILES, for pattern "G0 K1" and the y_{010} strategy at threshold, $t = 0.9$. In bold are the features for which differences were observed between the two cell types.

TF	Gene	w_k	p-value	Organism
EZH2	ENSG00000177374	**2.003**	0.005	Mouse
EGR1	ENSG00000166025	$3.62 * 10^{-8}$	10^{-6}	Human
	ENSG00000165795	$1.89 * 10^{-8}$		
GATA2	ENSG00000166025	$8.9 * 10^{-9}$	0.01	Human
SUZ12	ENSG00000177374	$2.06 * 10^{-9}$	0.003	Mouse

Table 4. TFs that are likely responsible for gene expression, by statistical enrichment analysis and their corresponding, sorted, MILES weights.

are characterized as TFs in ChEA (Lachmann *et al.*, 2010) and elsewhere.

Regarding classification, further experimentation with the classifiers is needed. The results of MILES were investigated thoroughly, but not so for the other classifiers. For example, Simple MIL uses a soft labelling scheme, by selecting the upper 1% of the instances with highest prior probability and then applying the label of those to the bag label. It would be interesting to see if the instances that are picked as significant from Simple MIL agree with the ones from MILES. Another point, would be to change the internal implementation of MILES in the MIL Toolbox (Tax, 2013) and instead of using an SVM, use the implementation of the SLEP package, Sparse Learning with Efficient Projections (Liu *et al.*, 2009), to find concepts. Finally, other MIL classifiers could also be tested, such as Diverse Density (Maron and Lozano-Pérez, 1998), and their output be compared with that of MILES.

The labelling schemes were a device used to generate appropriate gene labels to use for classification. Although the classifier performance was high, a more adequate scheme is needed. Judging from the differences in performance, between the two strategies, the increased number of positive bags for the first strategy helps classification. That is a weak point of the y_{010} strategy, as a skin cell must not be the most adequate reference when comparing to two blood cells and a stem cell. Instead of comparing the 3 cell types of Tier 1 to a skin cell, for the classification of pattern "G0 K1", the two cell types could be compared to each other to generate the bag labels. Furthermore, the meaning of the labels could be changed entirely. A bag label could be generated by counting the numbers of TFs that are bound and not bound in the promoter region of a gene. If more are bound, then the bag is labelled with 1 and 0 otherwise. The instance labels would then signify if a TF is bound or not. This is, of course, existing information from the ChIP-seq experiments, but it could be used as a validation platform for the evaluation of the current findings.

Further testing would also be beneficial. Other small datasets of known pathways could be used to find if the performance is consistent in many scenarios. This is judged important, as for the WNT pathway – although it is well-studied from a biological point of view – there is not an up-to-date database where all information is available to be used for bioinformatics analysis. Furthermore, it would be interesting to test between the other combinations of cell types for other patterns of expression. The pattern "G0 K1" was deemed important, as it signified which genes change expression patterns due to cancer. Other expression patterns could also be interesting to study. Finally, testing with the genes for which gene expression is unknown, due to incompleteness of the ENCODE data, would be a real-life scenario worth investigating.

As a last remark, some theoretical aspects of MIL could also be tested for the classification scenarios of this study. One of these aspects is the increasing bag size. It is natural to expect that a larger bag size makes a problem harder to solve, increasing its

complexity (Babenko, 2008). But this may not always be the case, as more TFs binding to the promoter region of a gene may be relevant for gene expression and be helpful in classification. One way to test this would be to average the instances for every bag, thus representing every gene with a single feature vector. Then, using the number of instances as an extra feature in the new vectors, normal classification could be performed. Finally, a second aspect, related to the first one, could be to try to identify how many TFs are actually responsible, or needed, for gene expression, from a MIL point of view. The assumption that one positive instance can be enough to label a bag positive, may not always be true. An example of this, is how many image segments – of lava, ash clouds and a conical shape – one would need to call an image one of an active volcano (Cheplygina *et al.*, 2013). Similarly here, many TFs may be responsible together for gene expression and only when identifying all of these, can a bag be labelled positive. To test this, different methods of instance concatenation could be used, to describe the bags in a (dis)similarity space, thus investigating the informativeness of instances and determining a number for meaningful classification.

REFERENCES

Babenko, B. (2008). Multiple instance learning: algorithms and applications. *CrossRef PubMed/NCBI Google Scholar*.

Boulay, G., Dubuissez, M., Van Rechem, C., Forget, A., Helin, K., Ayrault, O., and Leprince, D. (2012). Hypermethylated in cancer 1 (hic1) recruits polycomb repressive complex 2 (prc2) to a subset of its target genes through interaction with human polycomb-like (hpcl) proteins. *Journal of Biological Chemistry*, **287**(13), 10509–10524.

Chen, Y., Bi, J., and Wang, J. (2006). MILES: multiple-instance learning via embedded instance selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**(12), 1931–1947.

Cheplygina, V., Tax, D. M., and Loog, M. (2013). Combining instance information to classify bags. In *Multiple Classifier Systems*, pages 13–24. Springer Berlin Heidelberg.

Consortium, T. E. P. (2004). The encode (encyclopedia of dna elements) project. *Science*, **306**(5696), 636–640.

Dietterich, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, **89**(1-2), 31–71.

Eksi, R., Li, H.-D., Menon, R., Wen, Y., Omenn, G. S., Kretzler, M., and Guan, Y. (2013). Systematically differentiating functions for alternatively spliced isoforms through integrating rna-seq data. *PLoS Comput Biol*, **9**(11), e1003314.

Kandoth, C., McLellan, M. D., Vandin, F., Ye, K., Niu, B., Lu, C., Xie, M., Zhang, Q., McMichael, J. F., Wyczalkowski, M. A., Leiserson, M. D. M., Miller, C. A., Welch, J. S., Walter, M. J., Wendl, M. C., Ley, T. J., Wilson, R. K., Raphael, B. J., and Ding, L. (2013). Mutational landscape and significance across 12 major cancer types. *Nature*, **502**(7471), 333–339.

Kooistra, S. M. and Helin, K. (2012). Molecular mechanisms and potential functions of histone demethylases. *Nature Reviews Molecular Cell Biology*.

Lachmann, A., Xu, H., Krishnan, J., Berger, S. I., Mazloom, A. R., and Ma'ayan, A. (2010). Chea: transcription factor regulation inferred from integrating genome-wide chip-x experiments. *Bioinformatics*, **26**(19), 2438–2444.

Li, W., Kang, S., Liu, C.-C., Zhang, S., Shi, Y., Liu, Y., and Zhou, X. J. (2013). High-resolution functional annotation of human transcriptome: predicting isoform functions by a novel multiple instance-based label propagation method. *Nucleic Acids Research*.

Liu, J., Ji, S., and Ye, J. (2009). *SLEP: Sparse Learning with Efficient Projections*. Arizona State University.

Liu, Y. I., Chang, M. V., Li, H. E., Barolo, S., Chang, J. L., Blauwkamp, T. A., and Cadigan, K. M. (2008). The chromatin remodelers {ISWI} and {ACF1} directly repress wingless transcriptional targets. *Developmental Biology*, **323**(1), 41 – 52.

Maron, O. and Lozano-Pérez, T. (1998). A framework for multiple-instance learning. In *Proceedings of the 1997 Conference on Advances in Neural Information Processing Systems 10*, NIPS '97, pages 570–576, Cambridge, MA, USA. MIT Press.

Svedlund, J., Koskinen Edblom, S., Marquez, V. E., kerstrm, G., Bjrklund, P., and Westin, G. (2012). Hypermethylated in cancer 1 (*HIC1*), a tumor suppressor gene epigenetically deregulated in hyperparathyroid tumors by histone h3 lysine modification. *The Journal of Clinical Endocrinology & Metabolism*, **97**(7), E1307–E1315.

Tax, D. (2013). MIL, a Matlab toolbox for multiple instance learning. version 0.8.1.

Vaquerizas, J. M., Kummerfeld, S. K., Teichmann, S. A., and Luscombe, N. M. (2009). A census of human transcription factors: function, expression and evolution. *Nature reviews. Genetics*, **10**(4), 252–263. PMID: 19274049.

Wagner, G. P., Kin, K., and Lynch, V. J. (2012). Measurement of mRNA abundance using RNA-seq data: RPKM measure is inconsistent among samples. *Theory in Biosciences*, **131**(4), 281–285.

TU Delft Thesis Supplementary Material

May 28th 2014
Pages 1– 1

Using the Multiple Instance Learning framework to address differential regulation

Dimitrios Palachanis*

Pattern Recognition and Bioinformatics group† Delft University of Technology, Delft, The Netherlands

Defended on May 28th 2014

This document contains further explanations and additional figures which were deemed unnecessary for an essential understanding of the material but which may nevertheless be of interest to the reader.

S-1 DATA SOURCES

The data of the ENCODE Project consists of experiments done by different labs. According to the type of experiment, all the results are organized in data tracks and the combined tracks of all different labs are organized into super-tracks. This can be overwhelming when one is unfamiliar with the datasets. Here are provided all the links to the data used for the current study.

To associate TFBSs to genes, a gene annotation file was needed. One was downloaded from http://www.gencodegenes.org/releases/3c.html. It was an older one, but still valid as indicated in release history http://www.gencodegenes.org/releases/. This file was chosen, because of the use of this annotation by Caltech for RNA-seq.

For gene expression, the experiments of Cold Spring Harbor are more extensive, as were done for more cell types. Unfortunately, RPKM expression values are not provided for a lot of genes. For this reason, the dataset of Caltech was preferred, although done for less cell types. The *Genes Gencode v3c* were downloaded for the 3 cell types of Tier 1 from http://genome.ucsc.edu/cgi-bin/hgFileUi?db=hg19&g=wgEncodeCaltechRnaSeq.

The TFBS Uniform ChIP-seq peak data were downloaded from http://genome.ucsc.edu/cgi-bin/hgFileUi?db=hg19&g=wgEncodeAwgTfbsUniform#TRACK_HTML, in the form of *narrowPeak* files.

For DNA methylation (by Reduced Representation Bisulfite Seq) the *.bed* files from ENCODE/HudsonAlpha were downloaded from http://genome.ucsc.edu/cgi-bin/hgFileUi?db=hg19&g=wgEncodeHaibMethylRrbs.

For Open Chromatin, the DNa-seq data of UW/Duke were downloaded in the form of bigBed (*.bb*) files from ftp://ftp.ebi.ac.uk/pub/databases/ensembl/encode/integration_data_jan2011/byDataType/openchrom/jan2011/fdrPeaks/.

Finally, for Histone Modifications, the data of ENCODE/Broad was downloaded in the form of *narrowPeak* files from: http://genome.ucsc.edu/cgi-bin/hgTrackUi?db=hg19&g=wgEncodeBroadHistone.

S-2 NON NORMALIZATION OF RPKM VALUES

Here we give the same Figs. as Section 3.2, without the pre-processing step of RPKM normalization. The labelling of the bags for the test dataset of the WNT pathway changes very slightly for each strategy (Fig. SF-1). The experiments here (Figs. SF-2 and SF-3) were performed with $C = 0.005$, which was the best value when doing the parameter optimization experiments.

The highest performances for MILES per cell type can be seen in Table ST-1.

S-3 VARYING OF FP-FN PENALTY THRESHOLD C FOR ALL CELL TYPES

For the highest performing thresholds, θ, t, of every cell type the optimization experiments for parameter C were repeated to justify the selection of $C = 0.005$ as the optimal value (Fig. SF-4).

*to whom correspondence should be addressed

†Pattern Recognition and Bioinformatics: http://prb.tudelft.nl

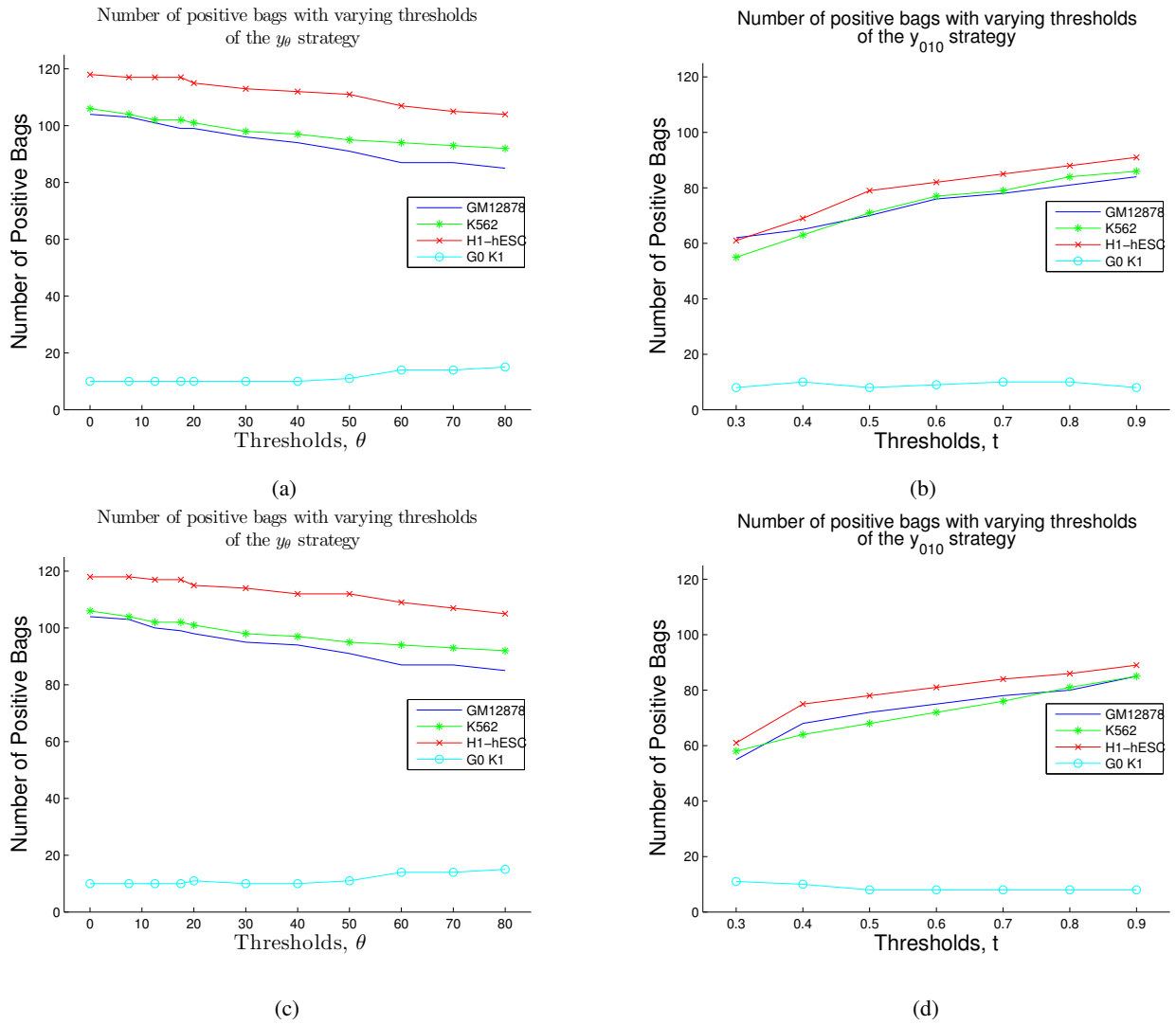
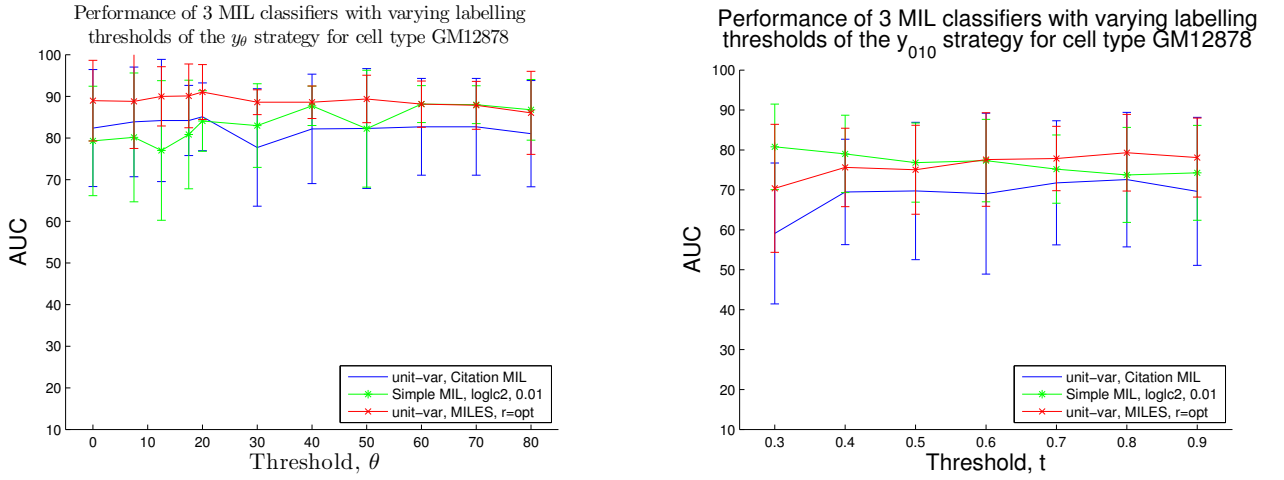


Fig. SF-1: **Number of positive bags to thresholds, for GO:0016055**: **With normalization**: **a**: Inverse proportionality to θ. **b**: Proportionality to t. **Without Normalization**: **c**: Inverse proportionality to θ. **d**: Proportionality to t.

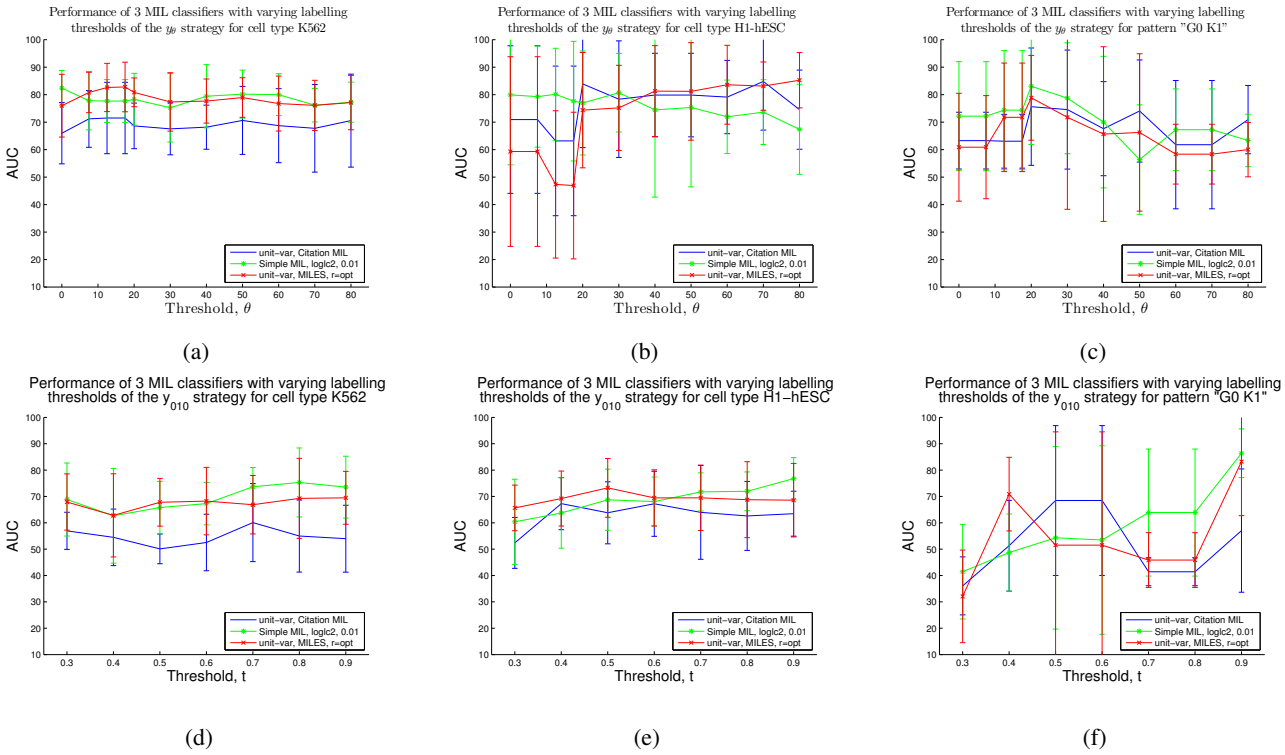
	Discretization Strategies			
	y_θ Strategy		y_{010} Strategy	
Cell Type	θ	AUC	t	AUC
GM12878	20	91.0% (6.6%)	0.8	79.3% (9.6%)
K562	17.5	82.8% (9.0%)	0.9	69.5% (10.0%)
H1-hESC	80	85.2% (10.1%)	0.5	73.2% (11.2%)
G0 K1	20	78.8% (15.4%)	0.9	83.2% (20.5%)

Table ST-1. Highest MILES performance for each cell type and corresponding thresholds, without RPKM normalization.



(a) (b)

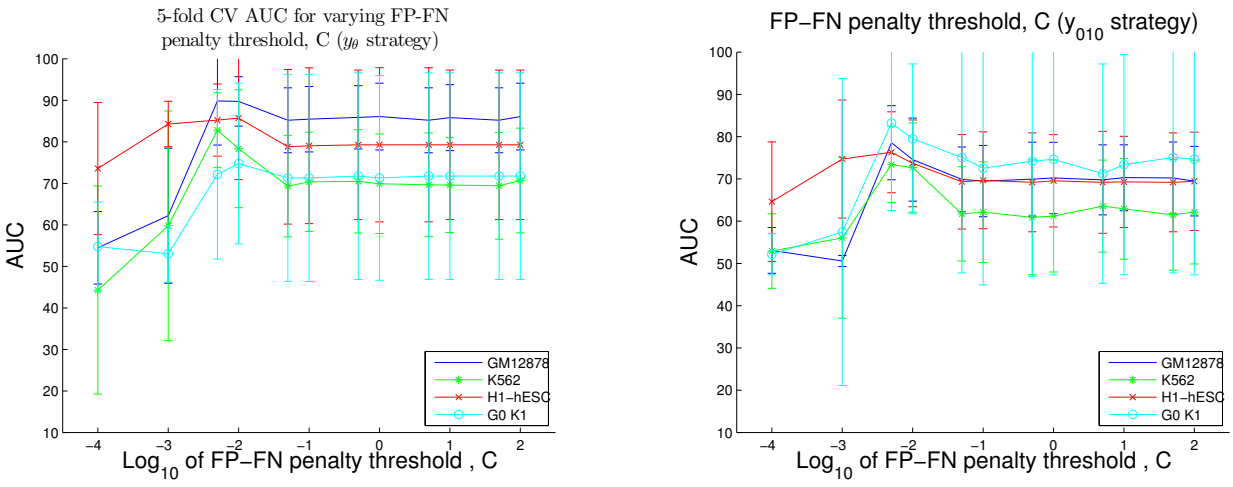
Fig. SF-2: **Classifier Performance without RPKM Normalization (GM12878)**: **a**: For the y_θ strategy. **b**: For the y_{010} strategy.



(a) (b) (c)

(d) (e) (f)

Fig. SF-3: **Classifier Performance for cell types K562, H1-hESC and pattern "G0 K1" without RPKM normalization**: **a**, **b**, **c**: For the y_θ strategy. **d**, **e**, **f** : For the y_{010} strategy.



(a) (b)

Fig. SF-4: **Varying of FP-FN penalty threshold C, for all cell types and the pattern "G0 K1"**: **a**: For the y_θ strategy. **b**: For the y_{010} strategy.

Work Log Document

Dimitrios Palachanis (0448850)
Leiden Institute of Advanced Computer Science
EWI, TU Delft

Contents

0.1 Motivation

Across multiple cell types, genes are expressed differentially and the question as to why that happens rises. It is of great importance to answer that question and find the regulatory elements that control gene expression across different cell types. A corresponding example in cancer research would be why some types of cells can be deregulated to have a cancerous expression and others do not. Accordingly, in this paradigm, one would be able to answer why a cell differentiates to a certain cell type, given only the regulatory elements of the genes involved in differentiation.

1 July 2013

1.1 A Brief Introduction

In the human genome there are about 20.000 genes. In every cell type, some genes are expressed – and some are not – and this is the reason why cells differentiate. For every gene (i.e. DNA sequence that is translated to a protein) there is a region upstream of the gene sequence, that enables RNA polymerase to bind to the DNA and initiate transcription, which is called a promoter region (PR).

Transcription factors (TFs) are proteins that bind to the DNA and control the flow of genetic information by activating or repressing the recruitment of RNA polymerase (Figure 1.1). Transcription factor binding sites (TFBSs) are DNA locations (sequences)

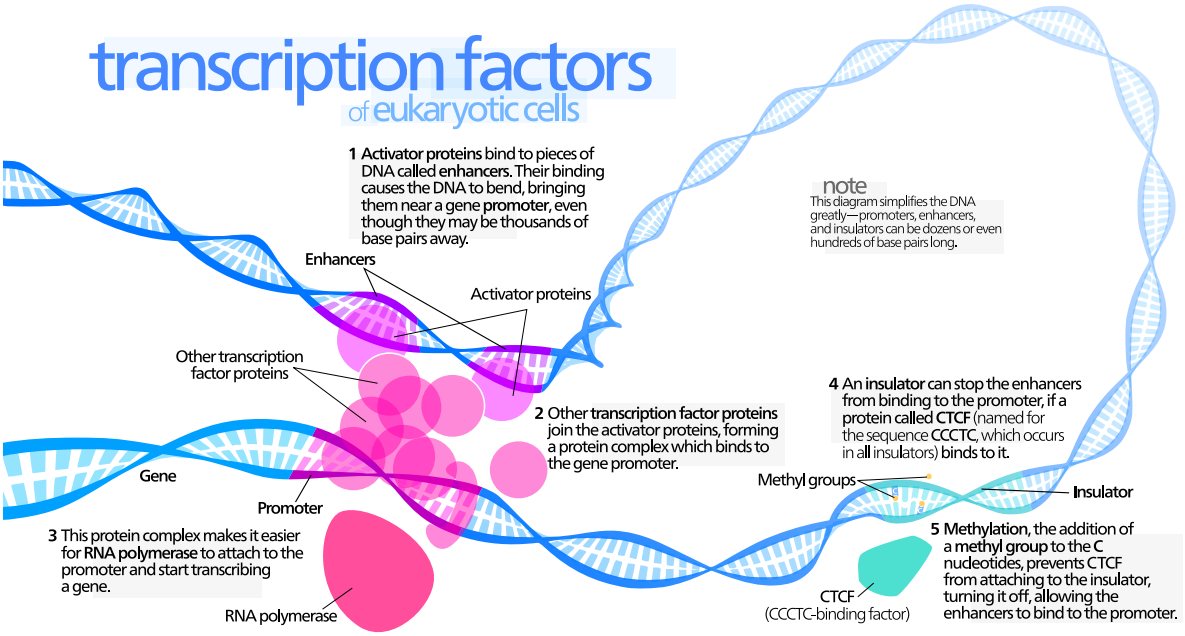


Figure 1.1: Illustration of DNA, TF and RNA polymerase binding. Taken from [1].

where proteins bind to the DNA (e.g. "Enhancers" label or "Insulator" label in Fig. 1.1). Promoter regions can be considered as a special case of TFBS. Promoter regions have the additional feature that RNA polymerase is the binding protein and the downstream sequence is translated to a protein.

The goal is to be able to predict if a gene will be expressed or not, given only the features characterizing the TFBSs and determining further which binding site(s) is/ are responsible for the expression pattern. In terms of machine learning and pattern recognition, this problem fits the criteria to be characterized as a multiple-instance learning (MIL) problem [2], where one object (bag) is described by multiple feature vectors (instances) but the label is only known for the object instead of the instances. In the current study, the object would be a gene and the instances would be the features of the TFBSs that are associated to its promoter region.

To solve this problem, data must be collected for every binding site associated to a gene PR from the ENCODE databases, to construct feature vectors (instances). Afterwards, the MILES (Multiple Instance Learning via Embedded Instance Selection) framework [3] will be used to train a classifier that will be able to predict the gene label (expressed/ not expressed) and which instances are responsible for this.

1.2 Data Collection

1.2.1 ChIP-seq Data

Data obtained from "Chromatin immunoprecipitation followed by sequencing" (ChIP-seq) [4] experiments can be used to detect DNA methylation, histone modifications or nucleosome distribution. But this data contains only the start and end position of a read for a specific TF in the form of two distributions. From all the short read alignments two distributions are generated. Two peaks form for the positive and negative strand flanking the binding location of the TF under study (Fig. 1.2). A peak-calling algorithm is then required to align the peaks to the exact position on the genome.

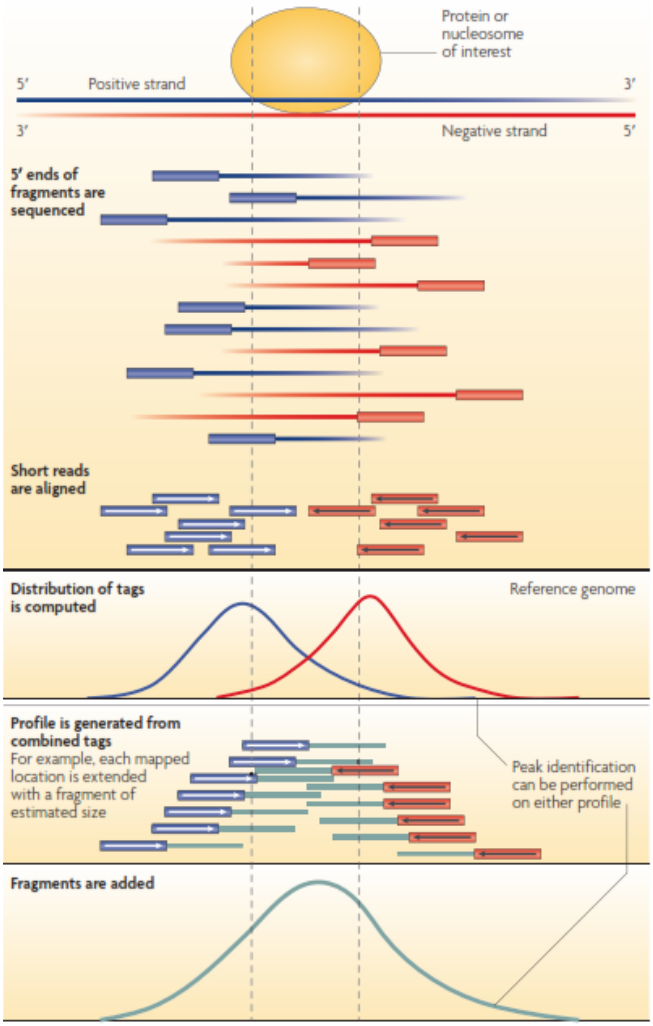


Figure 1.2: ChIP-seq data generation and preprocessing. The last two blocks correspond to a method for creating a single "profile" out of the 2 distributions, used to align the region of interest to the genome. Taken from [5].

In eukaryotes, the problem is that these reads may not be identical between cells of the same type. So, the two distributions may become very wide making the exact positioning of the TFBS on the genome impossible. If not impossible, then the position of a TFBS can be only accepted with a certain level of statistical significance.

1.2.2 Data Selection and Preprocessing

Cell types for the ENCODE Project are separated in 3 categories called Tiers. This is done for better data integration between groups. The 3 Tiers represent different priorities with Tier 1 being the highest. Priority is meant for experiments, as data should be collected from experiments done first for cells of Tier 1, then Tiers 2 and 3. Tier 1 consists of 3 cell types, GM12878, H1-hESC and K562, with the first two being normal cell types and the latter a cancerous cell type.

As of May 2013, there is a new browser track [1] that contains 690 datasets of transcription factor (TF) ChIP-seq peaks data. This track covers 161 unique TFs for 91 cell types. Though this looks promising, the problem is that the data is very sparse. A visual representation of this sparsity can be seen in the following Figure.

Figure 1.3: The sparsity of existing TF ChIP-seq data for the three cell types of Tier 1.

To collect data, a search for TFBSs was performed. Then, only the TFBSs that were associated to a gene were kept for further investigation (Fig. 1.4). To achieve this, all gene start positions were extracted from a gene annotation file and, from these, a list

[1] http://genome.ucsc.edu/cgi-bin/hgFileUi?db=hg19&g=wgEncodeAwgTfbsUniform#TRACK_HTML

of positions 1000 base pairs (bp) upstream were recorded, that represented the PRs. The positions of TFBSs with recorded ChIP-seq peak data were cross-referenced with that list. If a TFBS fell into a PR, it was stored in a new list. These are the TFBSs of interest for which feature vectors will be constructed.

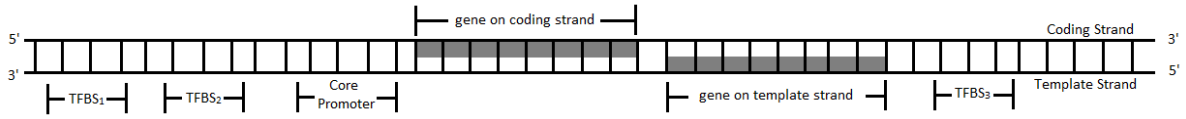


Figure 1.4: A toy example of interesting TFBS according to their position upstream of genes in the promoter region. $TFBS_1$ and $TFBS_2$ are associated with the gene on the coding strand. $TFBS_3$ is associated to the gene on the template strand.

For the preserved TFBSs, data such as ChIP-seq peaks, DNA methylation, histone modification and open chromatin will be collected to construct a "preliminary" feature vector (Figure 1.5).

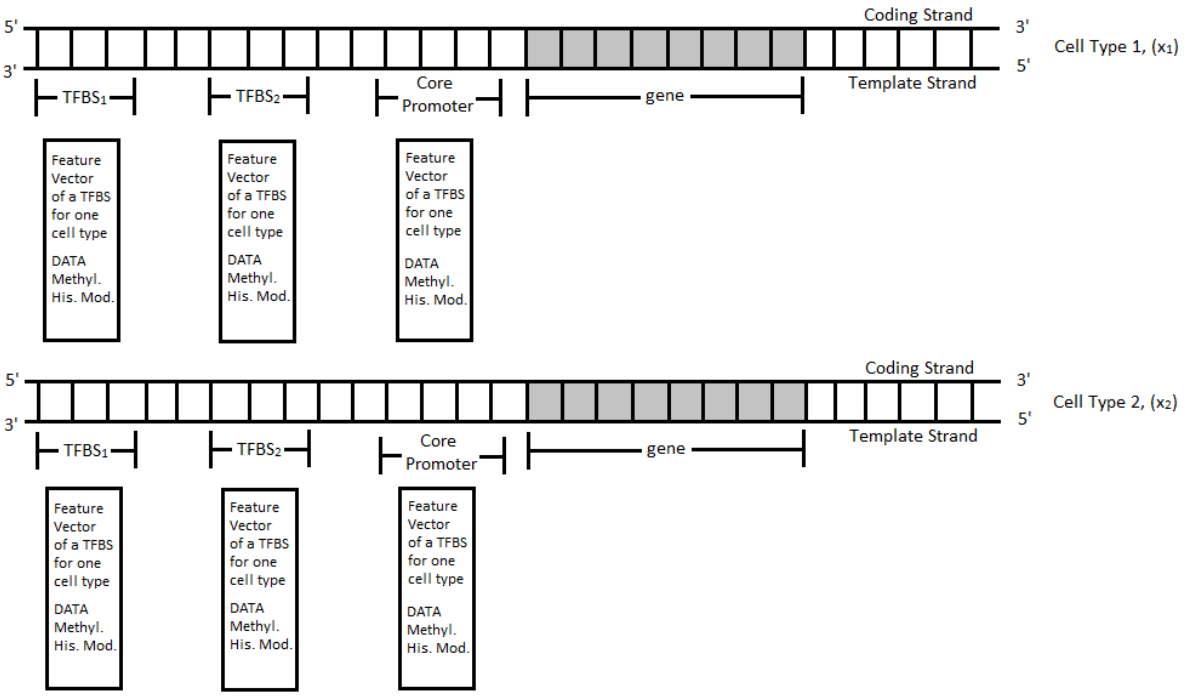


Figure 1.5: Constructing the "preliminary" feature vectors of the TFBSs for every cell type.

Finally, this will be done for all cell types and the resulting feature vectors will be concatenated to the feature vectors that will be used as the instances in the MIL framework.

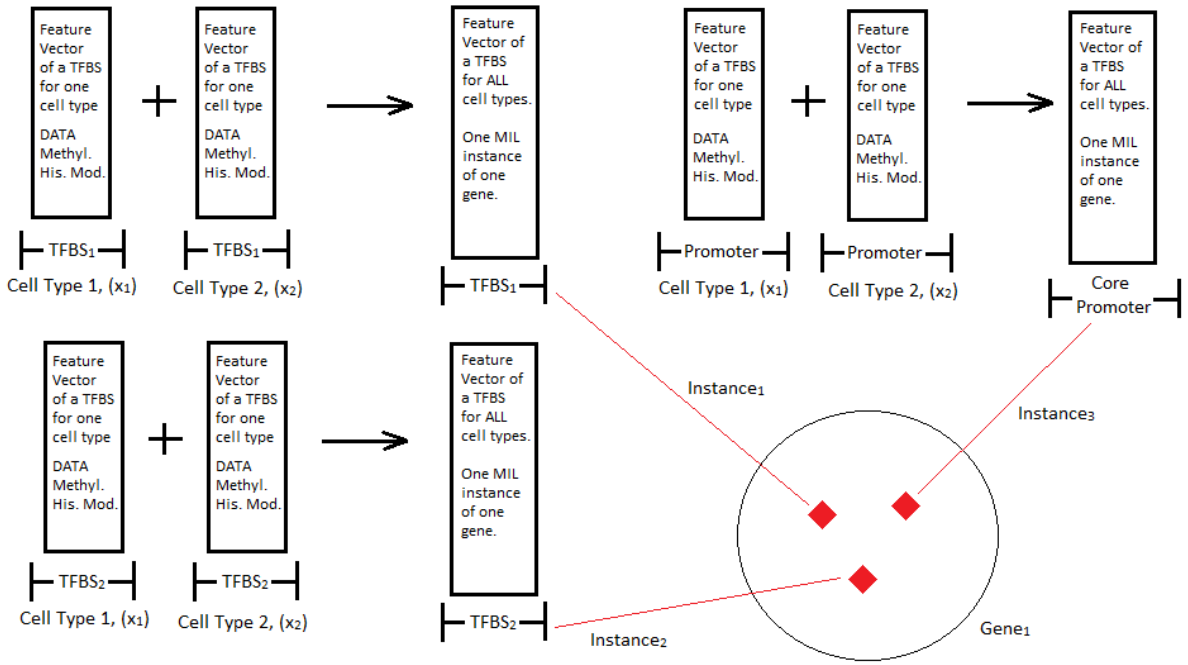


Figure 1.6: Instance and Bag construction in the MIL framework.

1.2.3 Visualizing the Data

After data collection and preprocessing the constructed dataset should look like Table 1.1. At present (July 2013), 284295 TFBSs have been assigned to 10055 genes out of 22565 only for one cell type (K562 of Tier1).

Cell Type	TF Name	Binding Site Features			Gene_ID	Expression
GM12878	Pol2	398	62.110	4.55		1
GM12878	Atf106325	1000	168.771	3.64		1
GM12878	Atf106325	677	106.042	3.64	ENSG00000229955	1
K562	Pol2	1000	355.866	4.55		0
K562	Atf106325	921	132.929	4.31		0
GM12878	Brf1	266	41.636	3.84	ENSG00000185238	0
K562	Brf1	843	26.308	3.84		1
GM12878	Cfos	144	22.658	3.85	ENSG00000028203	1
K562	Cfos	1000	242.247	3.85		0
K562	Pu1Pcr1x	131	20.579	4.19	ENSG00000270466	0

Table 1.1: The dataset constructed after the collection and preprocessing of data.

To enrich the data and obtain information for the rest of the genes, more cell types will be added. For the 10055 genes, the number of instances varies between 1 and 179. The number of instances per gene can be seen on Figure 1.7.

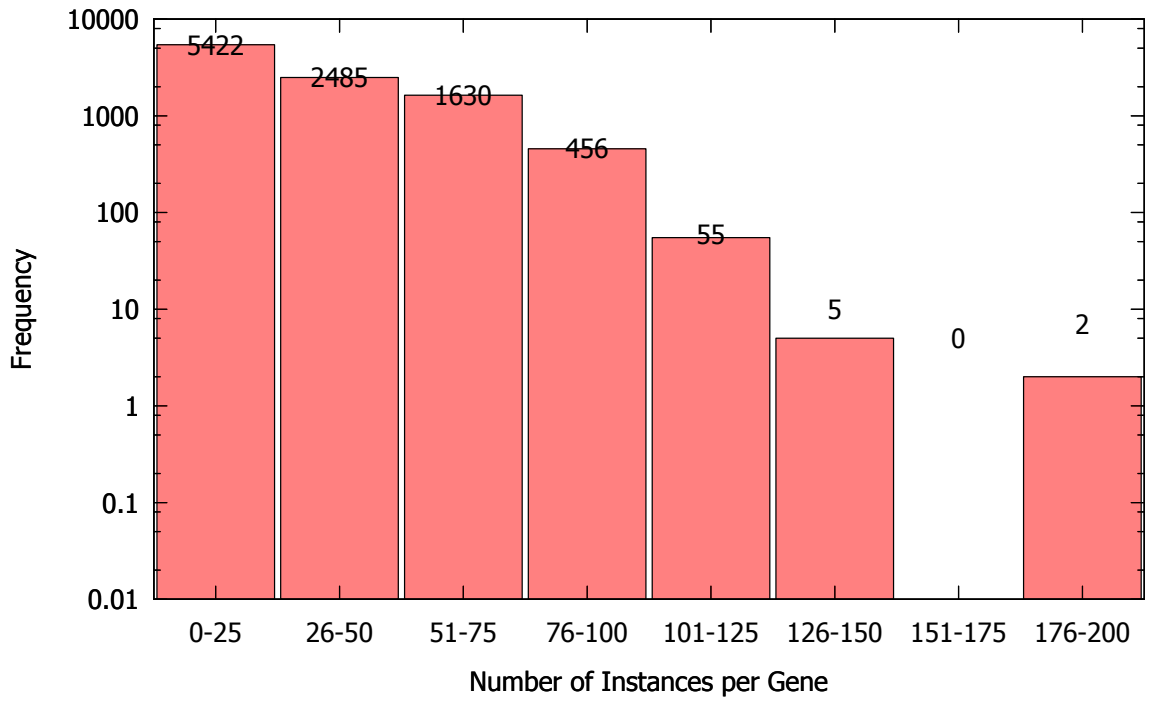


Figure 1.7: Frequency of the number of instances per gene. Most genes have 1-25 instances.

1.3 Using Multiple Instance Learning to Classify Gene Expression

In an MIL setting, each gene is considered as a bag, G_i, where $G \in \mathcal{G} = \{G_1, G_2, \ldots, G_N\}$ and $\mathcal{G}$ is the training set. Each bag label $y_i \in \mathcal{Y} = \{y_1, y_2, \ldots, y_n\}$, where y_i is a binary number. This label represents in how many types of cells the gene is considered as expressed or not. For example, for the three cell types of Tier1, a gene label could be "010", indicating that the gene is only expressed in cell type 2. This gives 2^n labels, where n is the number of genes. The bag labels are known, as they are determined from the biological data.

Each bag contains instances x_{ij}, such that $G_i = \{x_{i1}, x_{i2}, \ldots, x_{ij}\}$. Each instance, x_{ij}, contains biological information for a TFBS that lies inside a PR, across all cell types. The instance labels $y_{ij} \in \{0, 1\}$ are unknown. The instance labels express whether a certain TFBS contains (or does not contain) meaningful information for the expression of a particular gene.

The goal is, given a gene, to find the instances – the DNA locations where proteins bind in the promoter regions – that are responsible for the gene being expressed (or not) in a cell type. To classify gene expression and be able to find which instances are meaningful for it, MILES[2] will be used.

1.3.1 Instance-Based Feature Mapping

If N is the total number of bags, then l^+ is the number of positive ones and l^- is the number of negative ones and $N = l^+ + l^-$. Grouping all the instances together, disregarding whether they belong to positive or negative bags, the total number of instances is k and a random instance can be depicted as x^k.

Supposing the instances belong to an n-dimensional space, $\mathbb{R}^n$, the first goal is to try and map the data to a new instance-based feature space $\mathbb{F}_c$. That is needed, in order to be able to represents bags as single points.

To do this, a similarity measure is needed between an instance and a bag; and that is the shortest distance between an instance x^k and an instance x_{ij} in a bag G_i (Table 1.2) given as:

$$s(x^k, G_i) = \max_j \exp\left(-\frac{\|x_{ij} - x^k\|^2}{\sigma^2}\right) \tag{1.1}$$

A matrix can be constructed, containing all the distances between the instances and all bags. Each row of the matrix represents one of the features $s(x^k, \cdot)$ in $\mathbb{F}_c$ and each column is the coordinates $m(G_i)$ of a bag G_i in this feature space $\mathbb{F}_c$.

[2]Multiple-Instance Learning via Embedded instance Selection

$$\begin{bmatrix} s(x^1, G_1) & s(x^1, G_2) & \dots & s(x^1, G_n) \\ s(x^2, G_1) & \ddots & \dots & \vdots \\ \vdots & \dots & \ddots & \vdots \\ s(x^k, G_1) & \dots & s(x^k, G_{n-1}) & s(x^k, G_n) \end{bmatrix}$$

Table 1.2: The constructed similarity matrix. Each cell is the shortest distance between instance k and bag n.

With the bags mapped onto $\mathbb{F}_c$, and represented as single points, a classifier can be trained to differentiate between them.

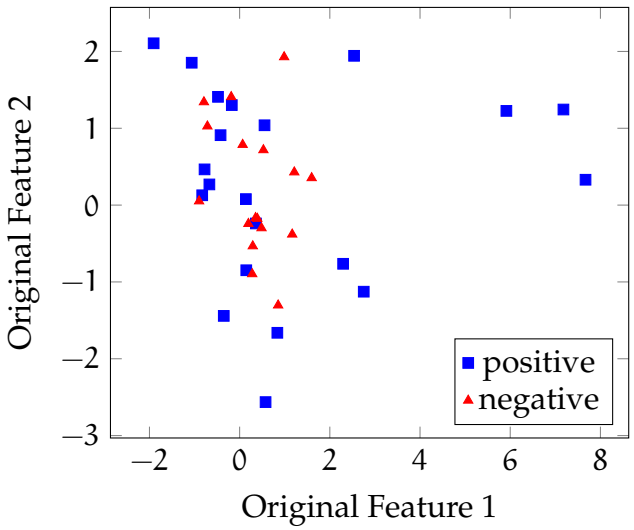


Figure 1.8: A toy example of 6 bags, 3 of which are positive and 3 negative, with 9,8,7,5,9 and 6 instances respectively (44 in total).

To perform feature selection and determine which instances are significant for the bag labels, a 1-norm support vector machine (SVM) will be applied and select rows from Matrix 1.2.

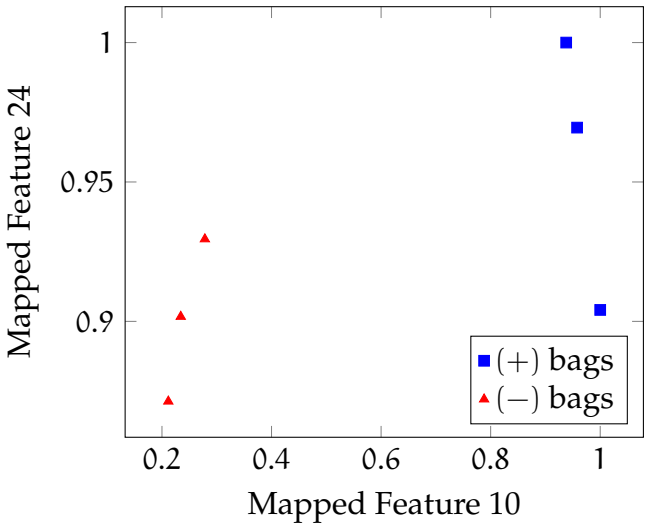


Figure 1.9: A toy example of bag mapping to concept feature space. Continuing the previous example, features 10 and 24 were selected from the SVM out of the 44. The 6 bags are mapped onto this 2D space.

The SVM tries to solve the problem of finding a linear classifier $y = sign(w^T m + b)$ in the new feature space $\mathbb{F}_c$, to differentiate between positive and negative bags. w and b are model parameters and $m \in \mathbb{F}_c$ are the bag coordinates in $\mathbb{F}_c$.

The weights of the linear classifier are restricted by demanding that: $y = w^T m^+ + b \geqslant +1$ for positive bags and $y = -(w^T m^- + b) \geqslant 1$ for negative bags. In standard SVMs the squared 2-norm of the weight vectors $\|w\|$ are used as a regularizer, which makes SVMs quadratic that are harder to solve. For this reason, MILES uses the 1-norm of w, $\|w\|_1 = \sum_k |w_k|$, which is linear and therefore easier to solve.

The classifier uses slacks ξ_i, η_j to account for possible overlap between the bags. The total error that must be minimized is: $C_1 \sum_{i=1}^{l^+} \xi_i + C_2 \sum_{j=1}^{l^-} \eta_j$, where C_1, C_2 are weights penalizing on false positives and false negatives. They are chosen such that they are convex, meaning if $C_1 = \mu$, then $C_2 = 1 - \mu$, where $0 < \mu < 1$. The 1-norm SVM is formulated as:

$$\begin{aligned} \min_{w,b,\xi,\eta} \lambda \sum_{k=1}^{n} |w_k| + C_1 \sum_{i=1}^{l^+} \xi_i + C_2 \sum_{j=1}^{l^-} \eta_j \\ s.t. (w^T m_i^+ + b) + \xi_i \geqslant +1, i = 1, \ldots, l^+, \\ -(w^T m_j^- + b) + \eta_j \geqslant +1, j = 1, \ldots, l^-, \\ \xi_i, \eta_j \geqslant 0, i = 1, \ldots, l^+, j = 1, \ldots, l^- \end{aligned} \tag{1.2}$$

To solve this in a linear programming way, w_k is rewritten as $w_k = u_k - v_k$, where $u_k, v_k \geqslant 0$. If either of them has to equal zero, then $|w_k| = u_k + v_k$. The 1-norm SVM

can be rewritten as:

$$\begin{aligned}
\min_{u,v,b,\xi,\eta} \lambda \sum_{k=1}^{n} (u_k + v_k) + \mu \sum_{i=1}^{l^+} \xi_i + (1-\mu) \sum_{j=1}^{l^-} \eta_j \\
\text{s.t.} [(u-v)^T m_i^+ + b] + \xi_i \geqslant +1, i = 1, \ldots, l^+, \\
-[(u-v)^T m_j^- + b] + \eta_j \geqslant +1, j = 1, \ldots, l^-, \\
u_k, v_k \geqslant 0, k = 1, \ldots, n, \\
\xi_i, \eta_j \geqslant 0, i = 1, \ldots, l^+, j = 1, \ldots, l^-
\end{aligned} \tag{1.3}$$

Any optimal solution to 1.3 will have at least one of the two variables u_k, v_k equal to zero for all $k = 1, \ldots, n$. If the optimal solution is $w^* = u^* - v^*$ and b^*, then the magnitude of w_k determines the influence of the k-th feature on the classifier. The index set of selected features is named $\mathcal{I} = \{k : |w_k^*| > 0\}$. Finally, a bag G_i is classified by:

$$y = \text{sign}\Big(\sum_{k \in \mathcal{I}} w_k^* s(x^k, G_i) + b^* \Big) \tag{1.4}$$

2 August 2013

2.1 Notes on Implementation

Looking ahead to the data to be added – DNA methylation, histone modification and open chromatin –, it became apparent that each TFBS will have a variable number of e.g. methylation sites associated to it. So this data cannot be incorporated directly to the feature vectors already constructed, because the dataset will end up having vectors of variable length. So, for now, the decision was made to just count how many sites are associated to each TFBS.

A toy example of the dataset already constructed, can be seen in the following Table.

Chromosome	Strand	TF	Start	End	Features	Gene_ID	Expression
7	-	Pol2(Ifng6h)	72936803	72937047	...	ENSG00000009954	0.3045
7	-	Pol2(Ifng6h)	72936408	72936590	...		

Table 2.1: A toy example of the TFBSs that need to be associated with other data.

And an example of methylation data to be added:

Chromosome	Strand	Start	End	Reads	Percentage
1	+	1000170	1000171	46	35
1	-	1000206	1000207	53	26

Table 2.2: A toy example of methylation data to be associated to TFBSs.

The use of dictionaries in python seemed appropriate for this association task. Looking at the example data, a problem becomes obvious; that there is no value in this data that is immediately descriptive and unique to be used as a key. The naive approach was first tried, where for each TFBS, the whole methylation data would be cross-checked to see if any can be associated to that TFBS. This approach was very slow, as there are currently 279178 TFBSs and more than 110000 DNA methylation peaks.

A different approach was needed, where an effective key would be used. A list of tuples was created from the information of Table 2.1, `[(1,7,-,Pol2,72936803, 72937047),(2,7,-,Pol2,72936408,72936590),...]`, where the first element of each

tuple is the unique row number. Then, for every chromosome, a dictionary was constructed. For each dictionary, the range of each TFBS peak was broken down to single nucleotides and each was used as the key. So, for the toy example, the dictionary of chromosome 7, would have entries like: `72936803:[1,Pol2,-],72936804:[1,Pol2,-],...,72937047:[1,Pol2,-], 72936408:[2,Pol2,-], ...`. This is a one-to-one mapping that makes it faster to search for methylation peaks that fall within a TFBS. A pseudo-code version of this algorithm can be seen below.

Algorithm 1 Associate data to TFBSs.

```
TFBS_data = []
with open(dataset) as indata:
    data = read file
    for i,line in enumerate(data):
        get chromosome, start, end, strand
        value = (i, chromosome, start, end, strand)
        TFBS_data.append(value)

# Create Dictionary as well. The key is row number i.

# List holding all chromosome dictionaries
chrom_dicts = [23*{}]
for name, chrom, strand, start, end in TFBS_data:
    # The range of chromosome positions this TFBS overlaps
    position_range = xrange(start, end+1)
    # Add TFBS name to list that overlap each nucleotide
       position.
    for p in position_range:
        chrom_dicts.setdefault(p,[]).append(name)

# Create a list that will hold the sum of methylation values
    for each TFBS
TFBS_meth = [0]*len(BS_dict.keys());
with open(Methylation_data) as methyl:
    for line in methyl:
        get chromosome, position, strand
        if position in chrom_dicts[index]: # lookup position
            for TFBS_ID in chrom_dicts[index][start]: # for
               all TFBSs associated to that nucleotide...
                TFBS_meth[TFBS_ID] = TFBS_meth[TFBS_ID] + 1;
                    # ... add 1 for the open chromatin peak

# Create a concatenated list of the dictionary values and
   the number of methylation peaks
concat = [ BS_dict[kindex] + [TFBS_meth[kindex]] for kindex
   in xrange(len(TFBS_meth))]
```

With this method, DNA methylation and open chromatin peaks were associated to TFBSs. The frequencies of the number of peaks associated to each TFBS are given in

the following graphs.

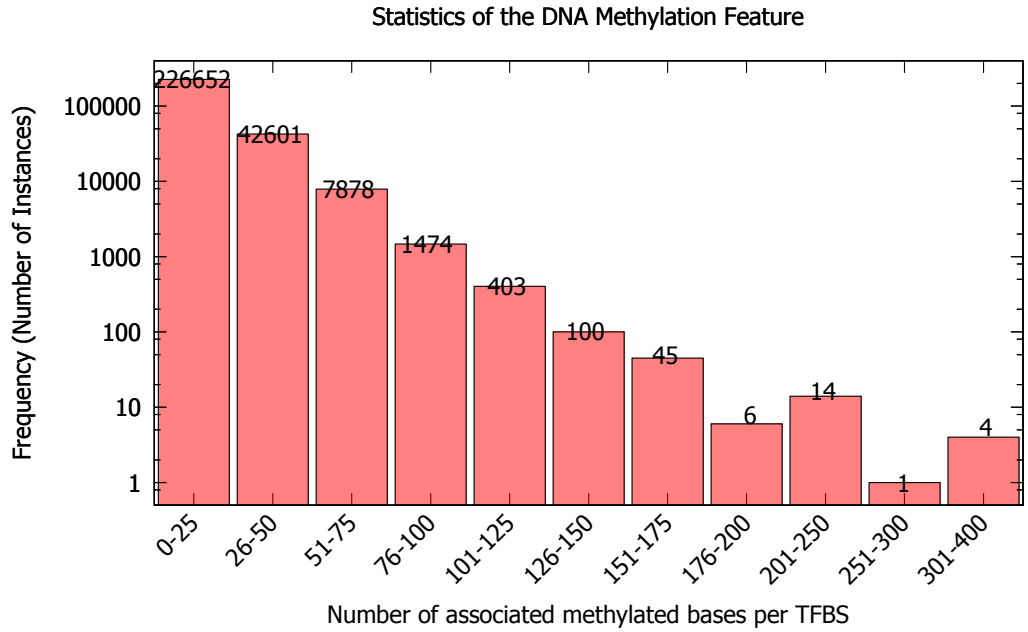


Figure 2.1: Frequency of the number of methylated DNA bases per TFBS. Most instances have 0-50 nucleotides associated to them.

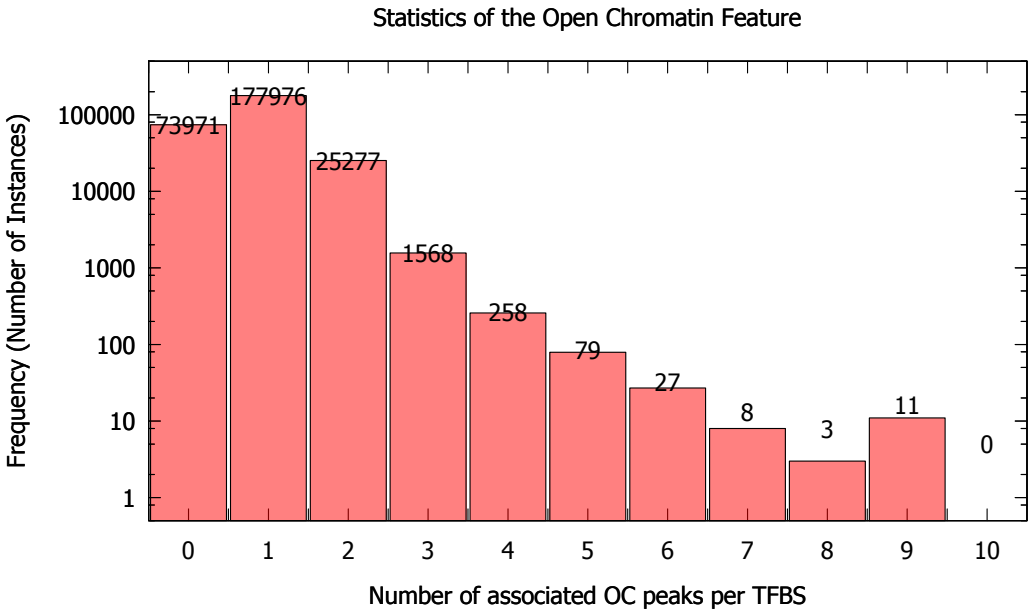


Figure 2.2: Frequency of the number of associated OC peaks per TFBS. Most instances have no or one peaks associated to them.

2.1.1 Collecting Histone Modification Data

Biological Information

Histones are proteins that assist in the tight packing of DNA [6]. Five major families of histones exist: H1/H5, H2A, H2B, H3 and H4. Histones H2A, H2B, H3 and H4 are known as the core histones, while histones H1 and H5 are known as the linker histones. Eight histones form an octamer called chromatin, around which DNA can wind (Figure 2.3).

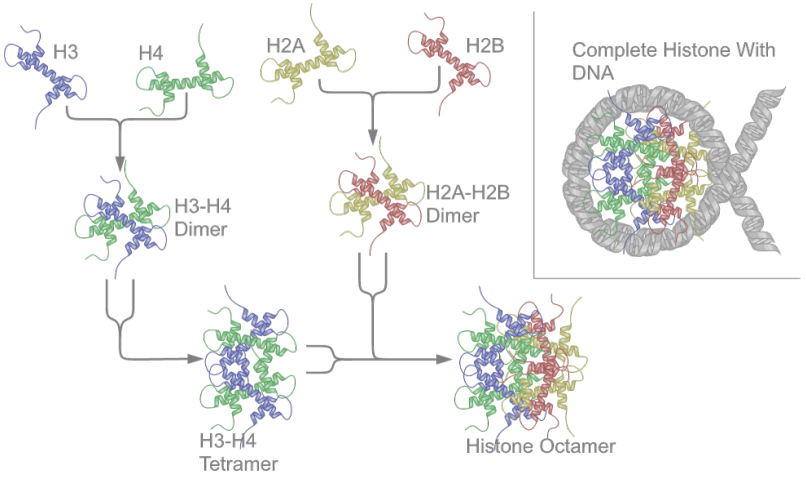


Figure 2.3: Chromatin forming from eight histones and the DNA packing around it. Taken from [6].

This complex formed between the chromatin and the DNA is called a nucleosome and is the first in a series of DNA packing mechanisms. Histones are interesting, as they play a role in gene regulation. From [7]: Genes, promoter regions or enhancer/-suppressor elements need to be accessible to fulfill their role during cell cycle. The composition of chromatin regulates the different genomic functions. But this composition is determined by histone modifications, such as methylation, so they are thought to play a role in cellular processes. The cause of histone modifications and their biological effects are debatable, but it is clear that different parts of the genome are associated with different patterns of histone modifications. These patterns were found with chromatin immunoprecipitation (ChIP) experiments using antibodies to specific histone modifications.

Histone methylation happens on one amine group of the Lysine amino acid. Lysine is repeated in multiple positions of the histones' sequence. According to which Lysine is methylated in the sequence, there are different association rules. In general:

1. Methylation of Lys4, Lys36 and Lys79 on histone H3 (H3K4, H3K36, H3K79) is associated with actively transcribed genes.

Trimethylated H3K4 is normally found at the promoter region or the transcription start site.

Trimethylated H3K36 is normally found in the gene bodies.

2. Methylation of Lys9, Lys27 on histone H3 (H3K9, H3K27) and Lys20 on H4 (H4K20) is associated with inactive genes.

3. On repressed genes, trimethylated H3K27 (H3K27me3) is associated with the promoters.

A schematic representation of this can be seen below:

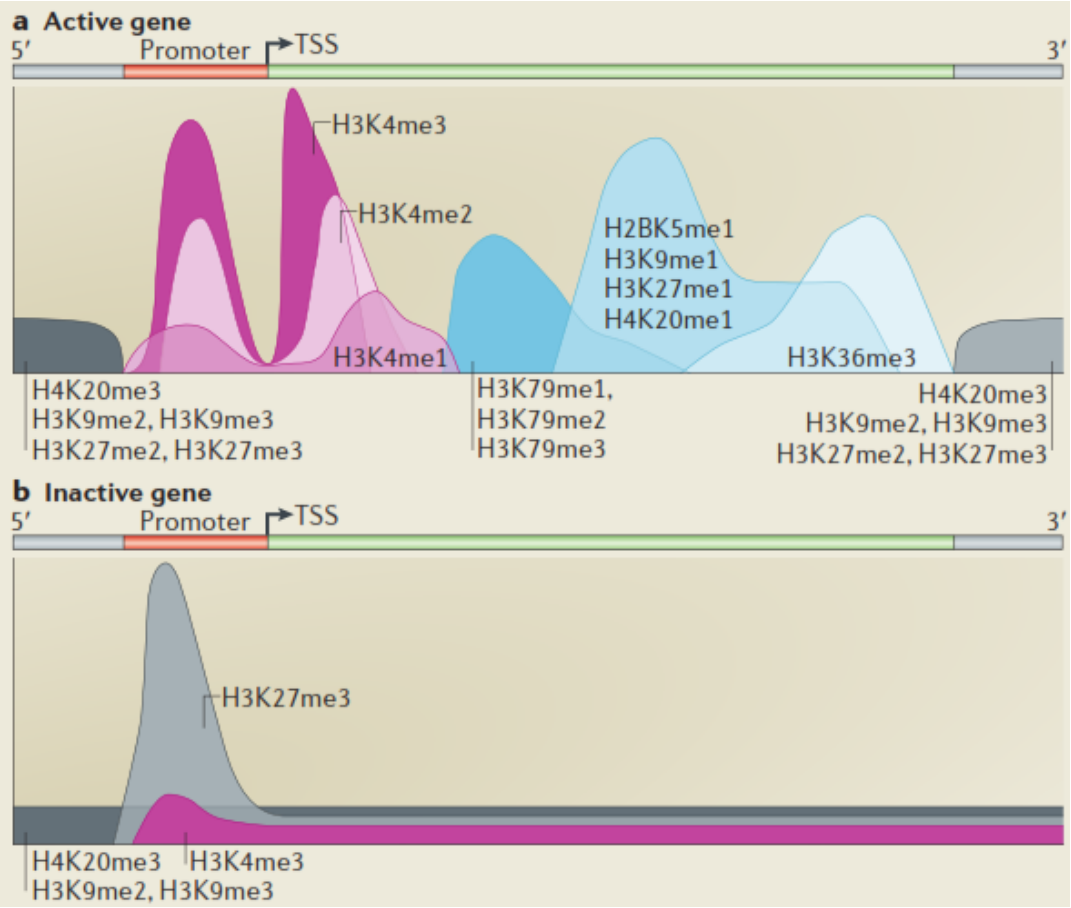


Figure 2.4: Average patterns of histone methylation on (a) actively transcribed and (b) inactive genes. Taken from [7].

The Datasets

[1]All ChIP-seq experiments were performed at least in duplicate, and were scored against an appropriate control. Submitted data was generally expected to meet an

[1]Taken from Histone Modification Data Description.

initial standard for inter-replicate consistency developed by the ENCODE Consortium to ensure an acceptable level of reproducibility; four fifths of the top 40% of the targets identified from one replicate (using an acceptable scoring method) should overlap the list of targets from the other replicate, or target lists scored using all available reads from each replicate should share more than 75% of targets in common.

Since every ENCODE dataset is represented by at least two biological replicate experiments, a measure of consistency of peak calling results was used between replicates, known as the irreproducible discovery rate (IDR) [8], in order to determine an optimal number of reproducible peaks.

Using this measure generated two datasets[2]; one called *conservative* and the other *optimal*. The optimal dataset contains some peaks that do not pass the original IDR threshold. The conservative dataset is a subset of the optimal and the one that was initially used for this study, as it contains a more confident set of peak calls.

The datasets selected were H3K4me3, H3K27me3 and H3K36me3. Two features were calculated for each TFBS; the first is the number of histone modification peaks that the TFBS falls into and the second is the distance from the "center" of the peak, calculated as:

$$distance = \frac{|\text{TFBS}_{start} - \text{HM}_{center}|}{\text{HM}_{width}} \tag{2.1}$$

The results will be shown in the following section, as substantial changes were made to the dataset.

[2]Taken from IDR Procedure for Histone Mark Datasets.

2.2 Revised Dataset

The initial method used to associate TFBSs to genes was computationally expensive. Once Python's `xrange` iterable was used to construct the features, the assignment of TFBSs to genes was redone to make the process faster.

In the narrowPeak files, there are almost 2 million ChIP-seq peaks. Of course, not all of these are going to be associated to gene promoter regions. Using the initial approach, approximately 284K binding sites were associated to 10K genes. Gene expression data was not available for all of these, so this dataset was somewhat reduced to 279K instances for 9.8K genes. With the revised method, 569K binding sites were associated to almost 17.3K genes. Again excluding genes for which there was no expression data, the dataset contains 561K binding sites for 16.9K genes.

As a first filtering step for both datasets, binding sites that do not belong to transcription factors were excluded. To achieve this a list of known Human TFs was collected [9] and if a binding site was in the list, it was assigned as a TF. For the initial dataset, 121K instances were recognized as TFBSs for 9K genes and for the revised dataset, 263K instances for 16.1K genes.

For the latter dataset, the statistics of the collected features are given in the following histograms.

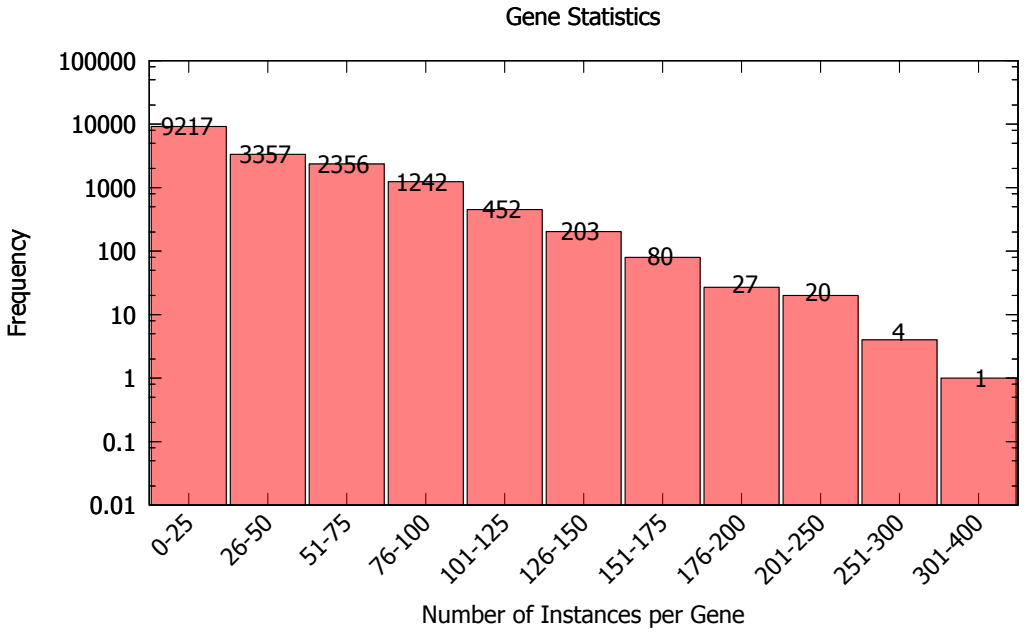


Figure 2.5: Frequency of the number of instances per gene. Most genes have 1-75 instances.

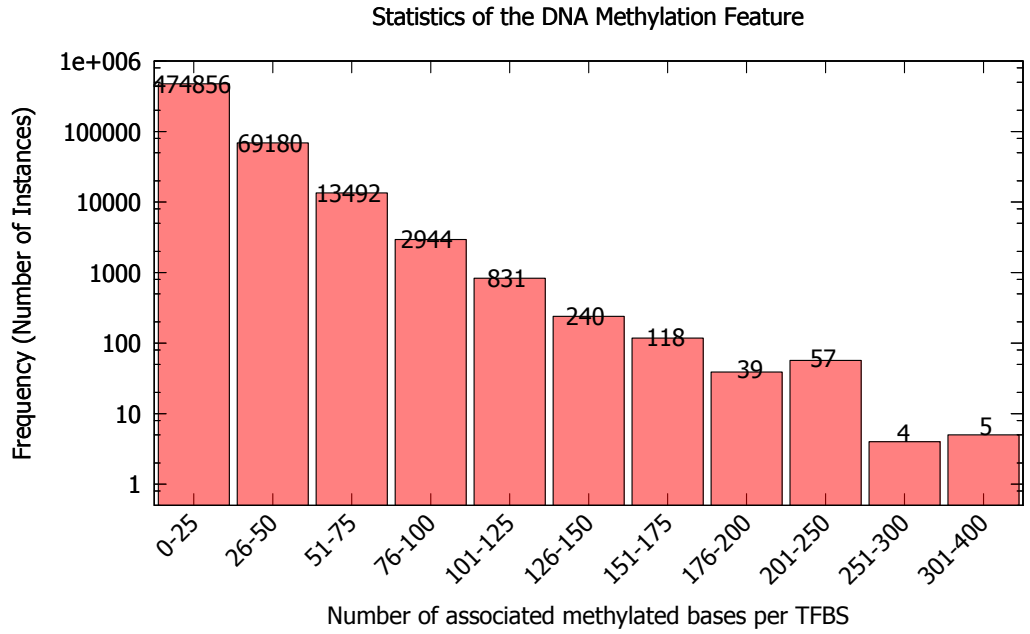


Figure 2.6: Frequency of the number of DNA methylated bases per TFBS. Most TFBSs have up to 50 methylated bases.

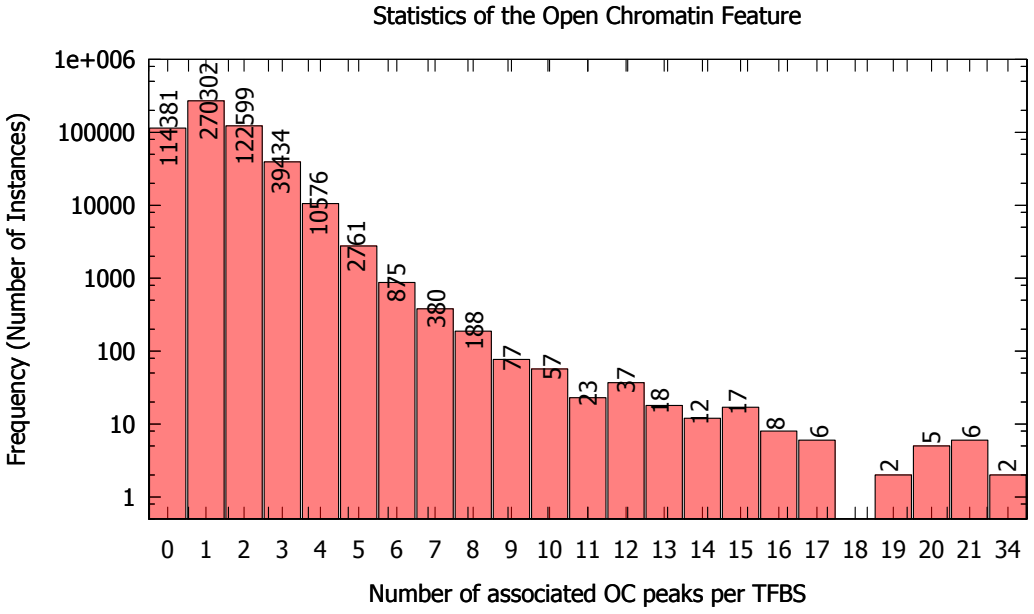


Figure 2.7: Frequency of the number of associated Open Chromatin peaks per TFBS. Most TFBSs have 0-3 peaks assigned to them.

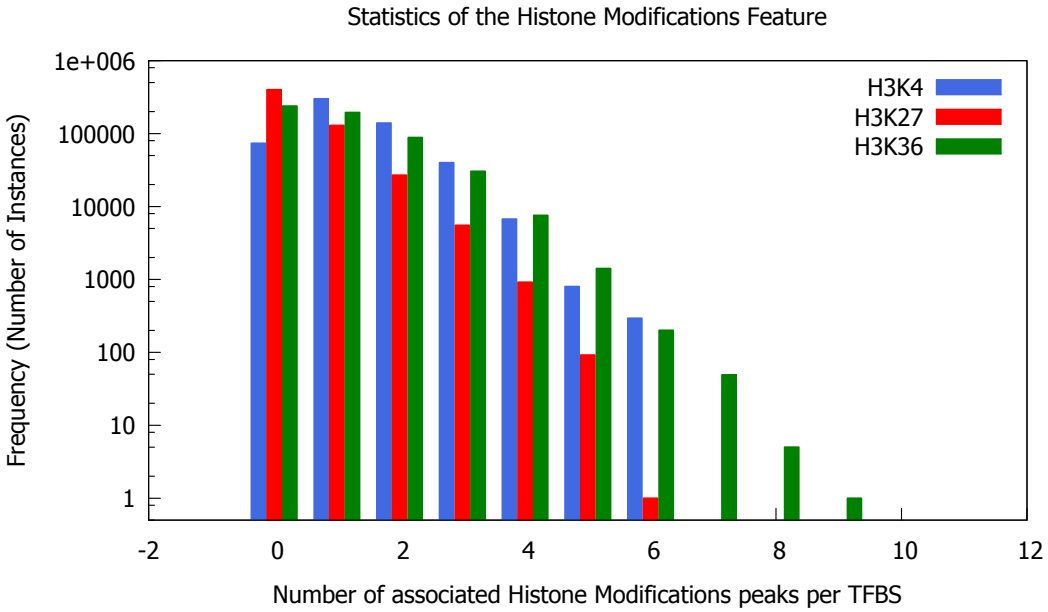


Figure 2.8: Frequency of the number of associated Histone Modification peaks per TFBS for Lys4, 27 and 36 on histone H3. Most TFBSs have 0-2 peaks associated to them.

A table containing the same data as Figure 2.8:

Number of Peaks	H3K4	H3K27	H3K36
0	73750	399184	238523
1	300692	129131	195053
2	139597	26923	88492
3	39909	5519	30468
4	6724	916	7560
5	800	92	1414
6	294	1	201
7	0	0	49
8	0	0	5
9	0	0	1
Sum	561766	561766	561766

Table 2.3: Frequencies of the number of associated Histone Modification peaks per TFBS.

3 September 2013

3.1 Dataset Re-revisited

The assigning of genes' promoter regions to ChIP-seq peaks was done in Python. The dictionary for every chromosome was held in a list. Initializing that list turned out to be a problem, as Python handles different methods in unique ways. Making a list of dictionaries with `[{}]*len(genes)`, uses the same reference to the inner list as the elements of the outer list. But this is problematic, as a gene would be assigned to all chromosomes that have the same relative position, instead of just one. This means that the end result would be having a gene assigned to chromosomes 1 and 17, if both contained a peak at position 10.000. To rectify this, a different initialization method, `[{} for i in xrange(len(genes))]`, is enough.

This changes the dataset yet again. It now consists of 345185 instances for 11200 genes. After appending gene expression data, the dataset is reduced to 341886 instances for 11035 genes. The procedure for appending DNA methylation, Histone modification and open chromatin data remained the same. After an initial filtering for binding proteins that are TFs, the dataset was finally reduced to 147564 instances for 10261 genes.

3.1.1 Pipeline Order

While constructing the latest dataset, an initial approach was to first filter for binding proteins that were actually TFs and then continue appending the rest of the features. This would lose information, as the data for non-TFs would be discarded. Another approach was to construct the complete dataset and then filter for TFs. Surprisingly, the end results, after filtering, were not the same. A bug in dictionary construction was found and corrected, as duplicates were generated by using the same files multiple times.

3.2 Removal of Duplicates

In the constructed dataset there are two kinds of duplicates. Experiments for the same binding protein from different institutes yield duplicate ChIP-seq peaks of the first kind. Experiments conducted for the same binding protein with varying experimental conditions yield duplicate peaks of the second kind.

The first step to remove them, was to bin them according to gene and binding protein. If the peaks were associated to the same gene and belonged to the same TF, then they were considered for duplicate removal. The second step was to use the q-values in the original narrowPeak files for every peak. The peak of lowest q-value was selected and the rest were considered as duplicates. Thus, the dataset was further reduced from 147564 instances to 103661.

3.3 Normalizing Data

For every ChIP-seq peak the center is given as an offset from the start. This integer value was translated to a float between 0-1 to better represent this feature. The other major alteration was to normalize gene expression.

To achieve this, the genes were binned according to their expression value. This resulted in the following histogram:

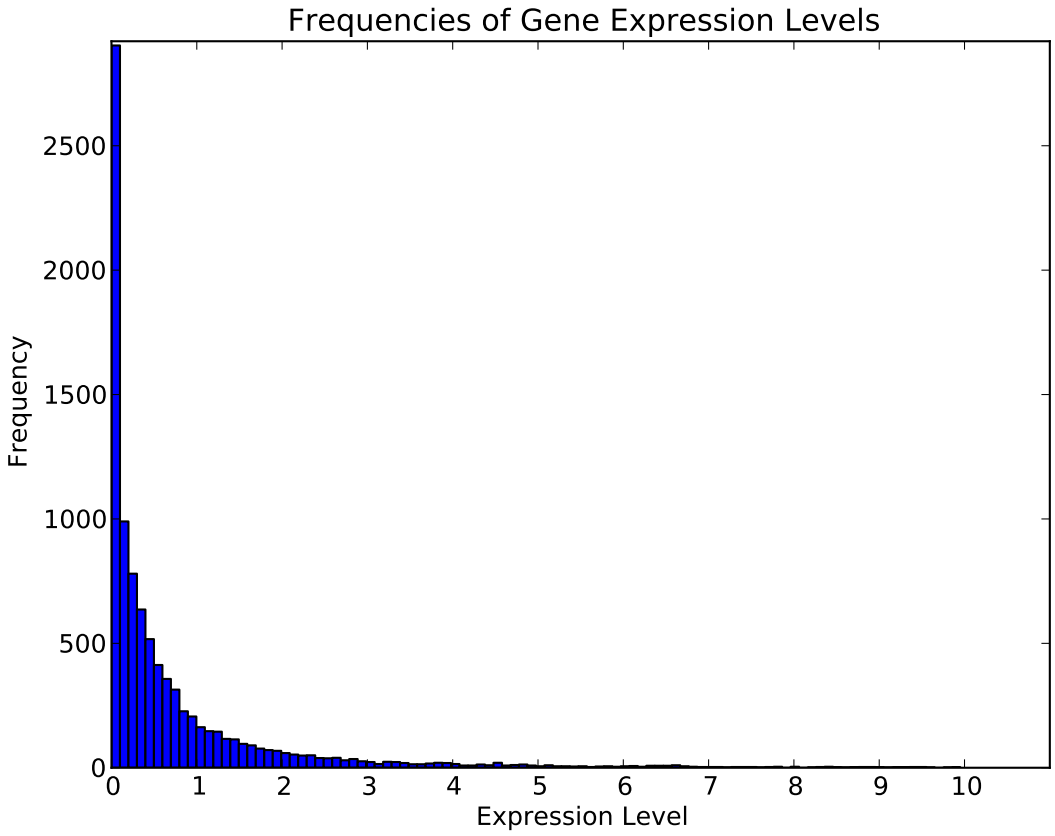


Figure 3.1: Frequencies of the expression levels of genes. Most of them are very lowly expressed.

Most genes are very lowly expressed; a threshold of 0.3 was set, below which all expressions were set to zero and to one above. This resulted in two quite balanced

classes of 5444 genes with 39566 instances belonging to the negative class (gene is off) and 4817 genes with 64095 instances belonging to the positive class (gene is on).

3.4 New Dataset Statistics

For the new, corrected dataset, the statistics of the different features are given in the following histograms.

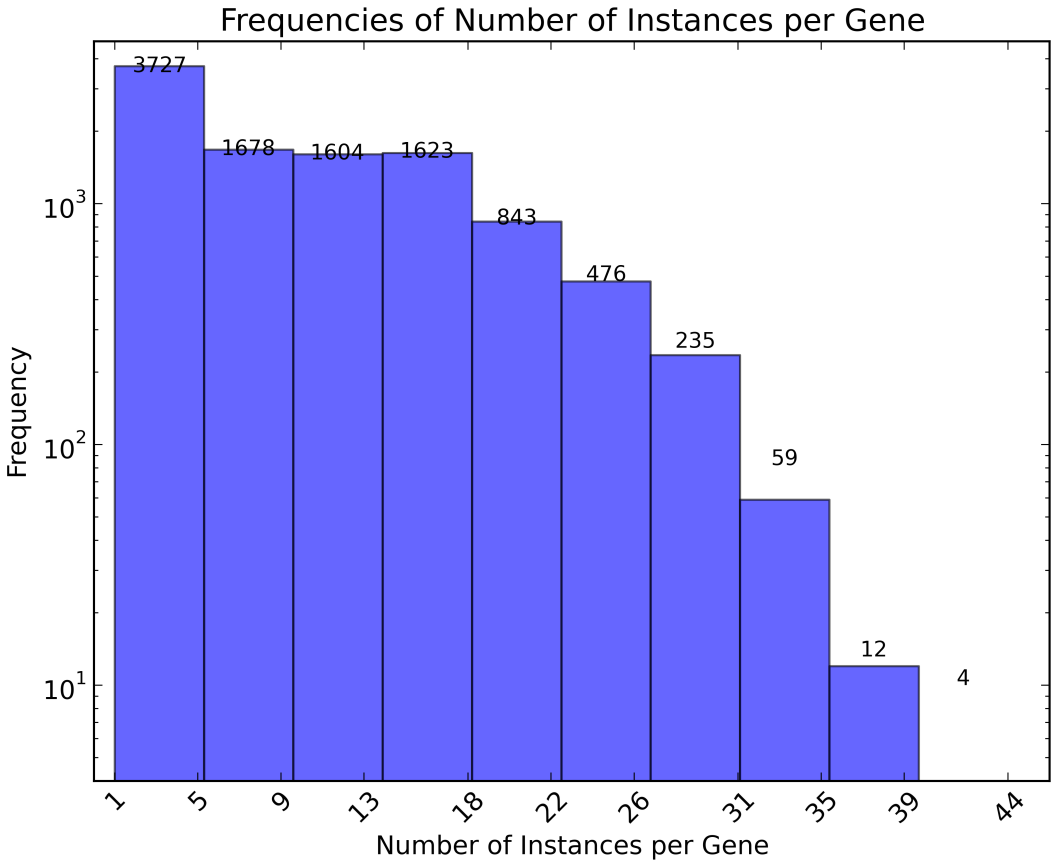


Figure 3.2: Frequency of the number of instances per gene. Most genes have 1-18 instances.

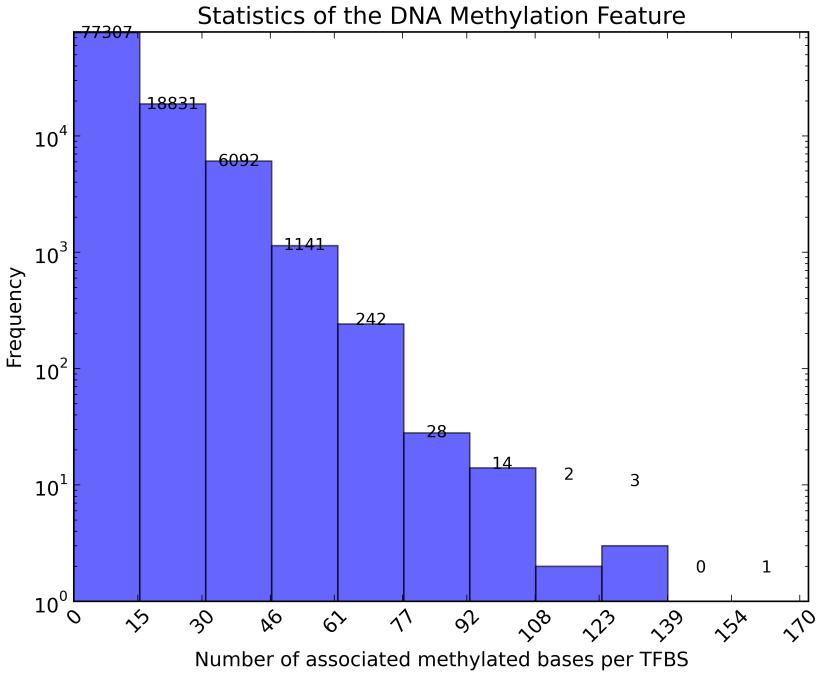


Figure 3.3: Frequency of the number of DNA methylated bases per TFBS. Most TFBSs have up to 50 methylated bases.

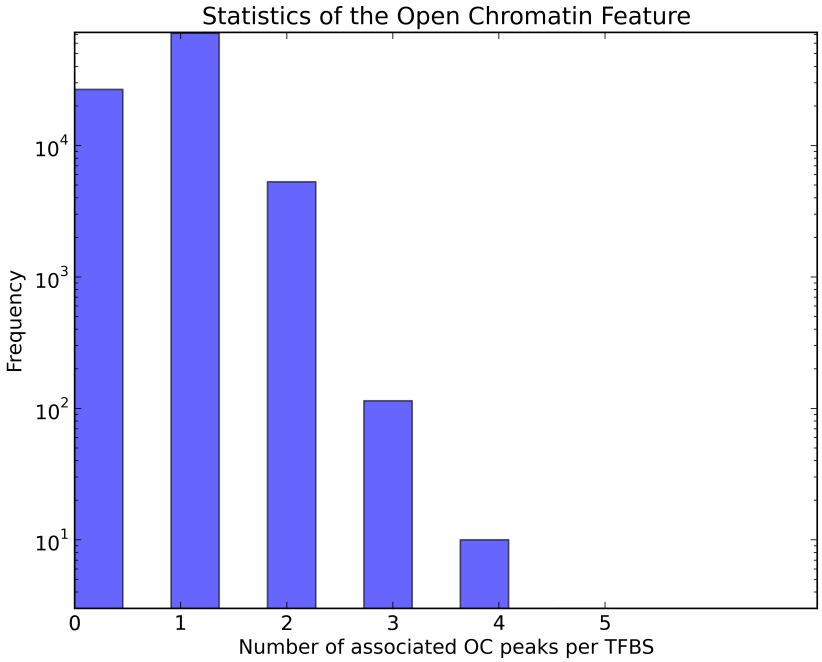


Figure 3.4: Frequency of the number of associated Open Chromatin peaks per TFBS. Most TFBSs have 0-1 peaks assigned to them.

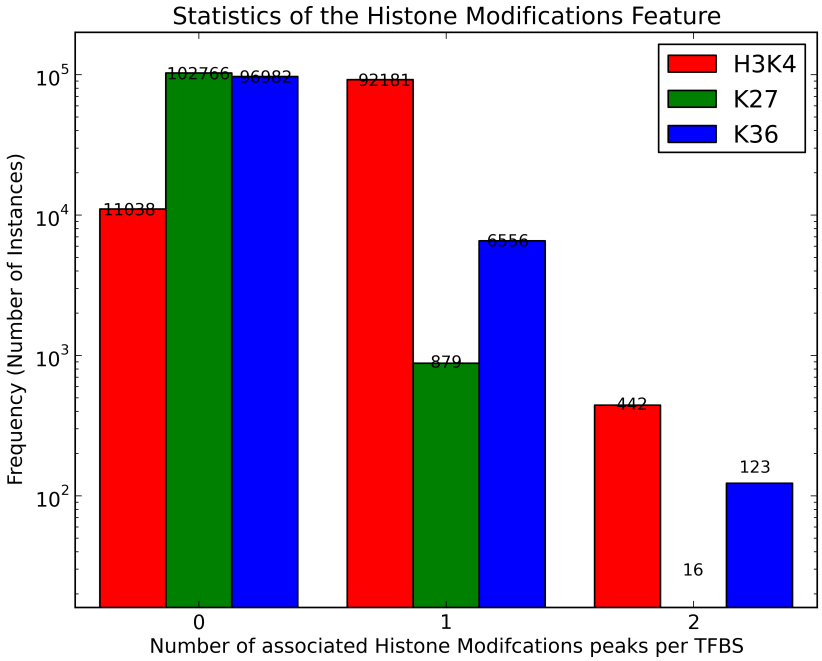


Figure 3.5: Frequency of the number of associated Histone Modification peaks per TFBS for Lys4, 27 and 36 on histone H3. Most TFBSs have 0-1 peaks associated to them.

Speculation...

Compared to the two previously constructed datasets, the new one has features that are deemed more meaningful biologically. Especially the number of instances per bag (gene), was reduced dramatically to a maximum of 45, which leaves roughly $^{1000}/_{40} = 25$ bp per binding protein. It is not known whether this is valid biologically, but it is seen as more correct compared to 400 instances per gene, previously.

4 October 2013

4.1 Dataset Construction Decisions

Recapitulating, to construct the dataset the following decisions were made:

1. A region of 1000 bp upstream of a gene was considered as the promoter region.
2. A binding site was considered to fall within a promoter region if its start, and not its end, would be there.
3. Only the binding proteins that are known transcription factors were kept in the dataset.
4. For the construction of some features certain datasets were used, e.g. for Histone modifications the *conservative* dataset was used.
5. Since different TFBSs can have, for example, a variable number of open chromatin peaks, the features constructed were just enumerating these attributes and not describing them, as this would result in vectors of variable length.
6. Gene expression was discretized to a binary feature at a threshold of 0.3, based on the frequencies of expression to produce balanced classes.
7. Some features were normalized.
8. Duplicate entries, meaning ChIP-seq peaks for the same transcription factor on the same gene, were removed by choosing the one with lowest q-value.

Since most features are binary (0/1) or between 0 and 1, having the positions of the TFBSs on the dataset is not optimal, as these are 6-7 orders of magnitude larger. The same stands for features such as DNA methylation that are 2 orders of magnitude larger.

4.2 Reconstructing the Dataset

The omission in all previous datasets was that each TFBS was described by features for one cell type. The correct way is to describe each TFBS by features of all available cell types, to incorporate the differences of this one position along cell types. This was

done by reconstructing the dataset and appending the available information from the 3 cell types of Tier 1, for every TFBS. The only difference is that when the dataset is cleaned, duplicates are considered among all cell types, so more data is going to be discarded.

4.3 Normalizing the Dataset

Since most features are binary (0/1) or between 0 and 1, the features that are not had to be normalized. To achieve this, the mean, μ, and standard deviation, σ of these features were calculated and the normalized feature would become: $feat_{Nor} = \frac{val-\mu}{\sigma}$. The threshold to discretize gene expression was set to 0.1, but this led to very imbalanced classes.

4.4 Matlab Experiments

All genes associated to GO term GO:0008284 (*Biological Process, activation of cell proliferation*), were isolated as a small example of the dataset, to import in MatLab. This subset contains 192 genes, of which 149 are negative and 43 are positive. There are 1350 instances in total.

The simplest possible setup was used:

- `Y,Z,I = milcrossval(A,5)`, to perform 5-fold cross validation. The only thing that this does is to generate a training set Y, that is 80% of the original dataset, and a test set Z, that is 20%. (*I have questions on that. Not working!*)
- `w = classifier(Y,conditions)` to train a classifier.
- `error = Z*w*testc`, to test.

The results can be seen on the following table:

Classifier	Result	Conditions
apr_mil	0.7027	(Y,'presence',0.1,0.99,0.0001,0.1)
maxDD_mil	0.5135	(Y, 'presence')
clust_mil	0.4116	(Y,1,4)
	0.4914	(Y,1,10)
	0.1595	(Y,1,100)
citation_mil	0.0531	(Y,1,1,3)
	0.0531	(Y,1,1,2)
simple_mil	0.0270	(Y,'presence',ldc)
MILES	0.1238	(Y,1,'r',1)
	0.1238	(Y,2,'r',1)

Table 4.1: Some initial (wrong) results!

5 November 2013

5.1 Discretizing Gene Expression

5.1.1 Introduction

Gene expression is measured with RNA-seq, a sequencing assay that can be used for quantification and transcript discovery. A common work flow of RNA-seq can be seen in Fig. 5.1.

The process starts by selecting a cell population and extracting total RNA. This term already implies that are different kinds of RNA that comprise the whole. Total RNA mostly consists of rRNA (ribosomal RNA) and other kinds such as mRNA (messenger), tRNA (transfer), lncRNA (long non-coding) and others. So, the first step is to select which kind of RNA will be sequenced. This is a subtractive process, meaning that the undesired kinds of RNA are depleted from the sample. The resulting sample is "enriched" in a particular RNA kind, so this step is called enrichment.

For gene expression polyadenylated mRNA is selected. Messenger RNA is the result of DNA transcription and is the molecule that contains the piece of biological information from DNA. RNA maturation is the process of adding a long sequence of adenines on the 3'-end of the RNA molecule, resulting in mRNA that has a poly(A)+ tail. This molecule can then leave the nucleus to be translated to a protein in the cytosol. mRNAs that are not matured are signified as poly(A)- and can include tRNAs, miRNAs (micro), piRNAs (piwi-interacting), siRNAs (small interfering), snRNAs (small nuclear) and lncRNAs.

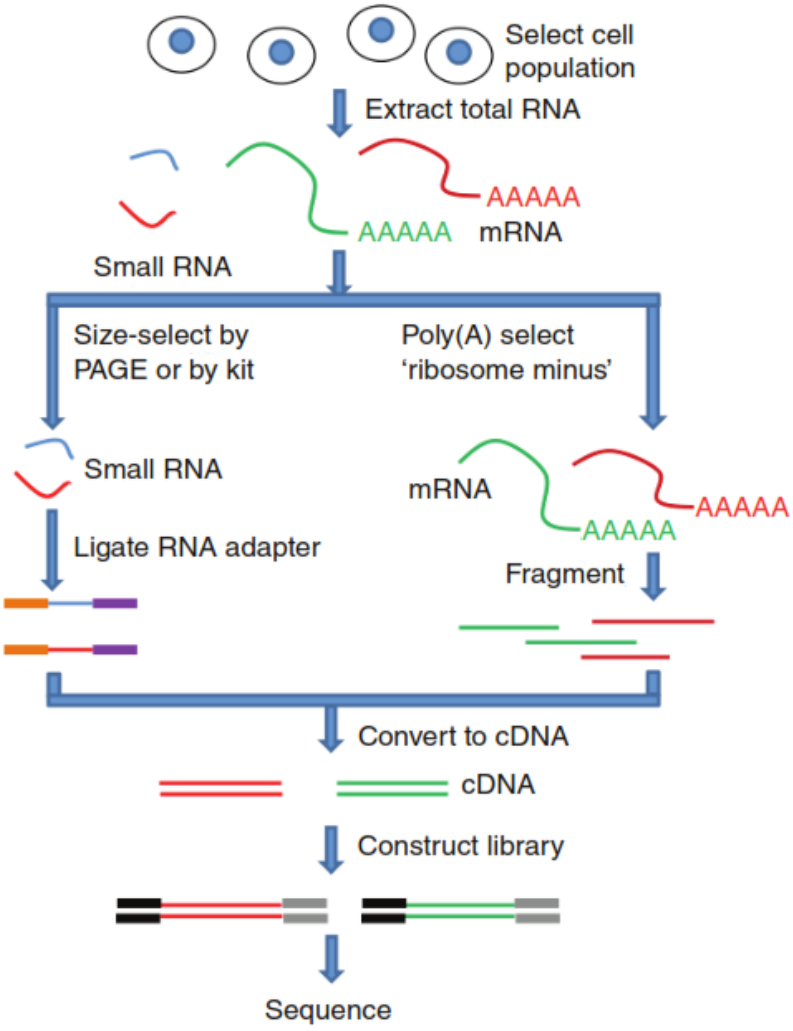


Figure 5.1: RNA-seq work flow (Taken from [10]).

After a sample is enriched for a certain RNA kind, the RNA molecules are fragmented evenly to ensure even coverage of whole transcripts, which means that every transcript must be encountered (read) roughly the same number of times [11]. After this is done RNA is reverse transcribed to complementary DNA (cDNA), so that it can be mapped to a reference sequence (e.g. known gene, exon sequence or whole genome). Finally, cDNA molecules are labeled with certain chemicals and thus, a library is constructed.

Gene expression values are usually represented with RPKMs, which stands for Reads per Kilobase per Million mapped reads. RPKMs are proportional to the abundance of each gene or transcript. This is a normalized value that corrects for the library size and reference sequence length [12]. It is given by the formula:

$$\mathrm{RPKM} = \frac{\mathrm{N}_{\mathrm{reads}}}{\mathrm{L}_{\mathrm{kb}} * \mathrm{N}_{\mathrm{reads}/_{\mathrm{million}}}} \tag{5.1}$$

5.1.2 Data Selection

The ENCODE Project contains RNA-seq data from 16 cell types, including the 3 of Tier1. These are separated in long (200 bp) poly(A)+ and poly(A)- experiments conducted with the standard RNA-seq protocol [13]. The poly(A)+ ones were selected for this study. Since all values for gene expression are given in RPKM units, they had to be discretized to a binary 0-1 value, to indicate if a gene is expressed or not.

5.1.3 Discretization Strategy

RPKM values are real, positive numbers that had to be descritized and used as gene labels for classification. To this end, a normalization step was first applied to the data. For every cell type, the RPKMs of all genes were added together. This sum represents the overall RNA production of a cell type. Each value was normalized by the mean of those sums:

$$N_c(g) = R_c(g)\frac{C}{\sum_{k\in G} R_t(k)} \tag{5.2a}$$

$$\text{where, } C = \frac{1}{|T|}\sum_{t\in T}\sum_{k\in G} R_t(k) \tag{5.2b}$$

where $t \in T$ are the different cell types in set T, with c a cell type of interest. g is the gene in consideration, over all possible genes $k \in G$.

This new value, $N_c(g)$, for a gene in a particular cell type, resembles the TPM (for Transcripts Per Million) values, that can be generated from RPKM, by dividing with the sum of RPKMs for all genes and multiplying by a million, that is stated to be a more accurate measure of RNA molar concentration [12].

After this preprocessing step, a list of 16 normalized expression values was created for every gene. The data of this list, was fitted to a log-normal distribution. There are two paths that can be considered here. One is to assume the values are already in log-space and fit to a log-normal distribution or to take the logarithm of the values and assume fitness to a normal distribution. The first path was considered, as the frequencies of expression, for every cell type, are equal to one per gene. Secondly, as a lot of genes have zero expressions for a certain cell type, using the logarithm can be problematic. Even compensating with a very negative value instead of 0, for the log, the resulting distribution is slightly shifted to the left.

After constructing this log-normal distribution, the second Pearson's skewness coefficient, $3(\mu - \mathrm{median})/\sigma$, was used to calculate which way, left or right, most of the probability lies. That in turn signifies a state for the gene, whether it is mostly off or on.

For a new gene expression value, the decision to discretize it as expressed or not works as follows:

1. If the skew is positive, most of the probability is relatively closer to zero, so the gene is assumed to be mostly not expressed. For a new value, if it falls under the fourth quantile, assign a label of 1, 0 otherwise.

2. If the skew is negative, most of the probability is relatively away from zero, so the gene is mostly expressed. If a new value falls under the first quantile, assign a label of 0, 1 otherwise.

This strategy can be seen in the following figure.

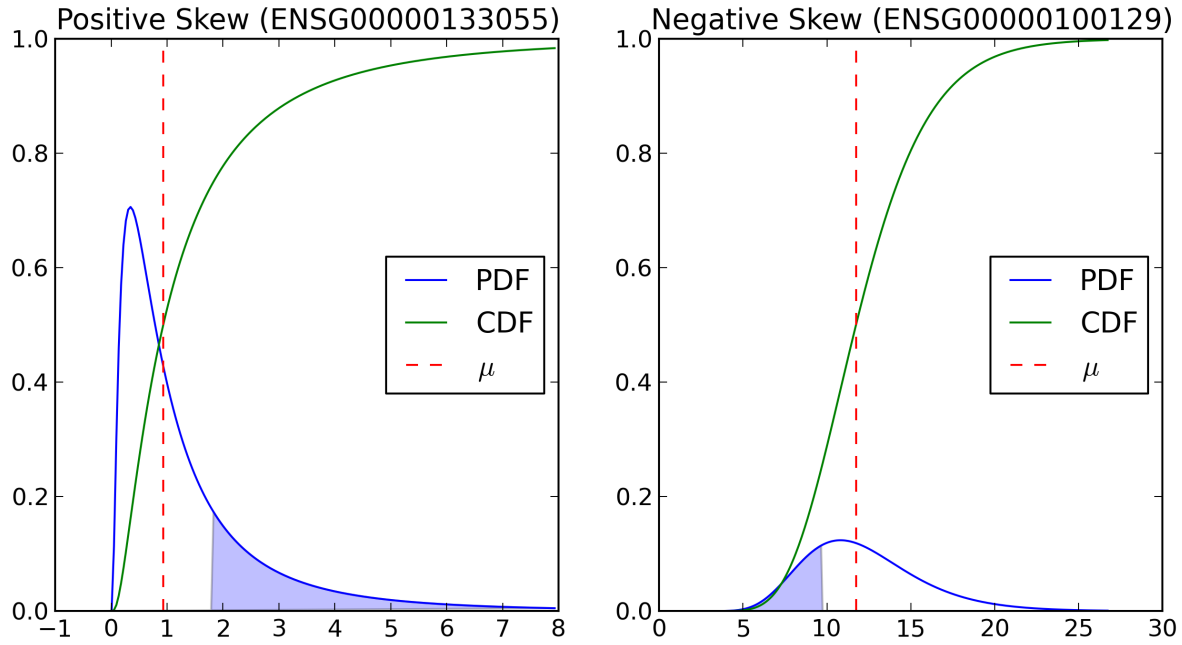


Figure 5.2: Inferred probability and cumulative distribution functions for two genes of the dataset with positive and negative skews. The decision boundaries (fourth and first quantiles) are shaded in light blue. Means, μ, are also depicted in red.

This method yielded the results seen in the following Table.

Cell Type	Positives Genes	Positive Genes across all 3 Cell Types
GM12878	3249	
H1-hESC	4356	419
K562	3221	

Table 5.1: Positive genes for the 3 cell types of Tier1 (out of 12570).

5.1.4 Matlab Experiments

With cross-validation working properly and with the new labels, the previous dataset changed slightly. Out of the 192 genes (bags) 42 are now positive and 150 negative. The previous experiment of a 5-fold cross-validation was repeated. The results are as follows.

	classifier		
error	Citation MIL	Simple MIL (Bayes-Normal-1, q=0.01)	MILES r=5.000000
AUC	**59.6 (10.7)**	**59.6 (11.0)**	**62.1 (12.7)**
cl.error	**28.7 (5.3)**	**27.6 (5.5)**	21.9 (1.1)

Table 5.2: MIL results of a 5-fold CV experiment for 3 classifiers. For citation MIL and MILES the variances of the features were scaled to one.

5.2 New Discretization Strategy

5.2.1 Additional Features

To hopefully increase classification performance, additional features were constructed from the existing data. Two important ones are genomic distances. One is the distance of each TF from the transcription start site (TSS) in bp and the other is the in-between neighbours between a certain TF and the TSS. Finally, the DNA methylation was expanded to look for bp that were methylated 50 and 100 bp away on either side of a TFBS.

5.2.2 Two Strategies

Since the previous method yielded poor results a new one was introduced and a previous one (used in Section 3.3) was reused. The first is to compare gene expression in the cell types of Tier1 to another, similar cell type. For this purpose, cell type CD20 of Tier2 was used, as it is a blood cell and relates to two out of the three cell types of Tier1; GM12878 and K562 are respectively a healthy and a cancerous blood cell. An initial threshold $t = 0.7$ was used. If a gene in one of the three cell types of Tier1 is lower than $t * R_{CD20}(g)$, then a 0 label is assigned, 1 otherwise. In this case, the labels do no longer represent the expression or non-expression of a gene, but the significant over- or under-expression in relation to cell type CD20.

The second, reused method is setting an arbitrary threshold, based on the frequencies of gene expressions (Figure 3.1). A threshold $t' = 0.007$ was used. The outputs of the two methods can be seen below.

	Positive Genes (out of 12570)	
Cell Type	Comparison to CD20($t = 0.7$)	Arbitrary Threshold ($t' = 0.007$)
GM12878	6583	8995
K562	6935	8954
H1-hESC	8591	10709

Table 5.3: Output labels of the two different discretization schemes.

With the new labels in place, two sets of experiments were designed, one per labelling scheme.

5.2.3 Evaluating Strategies

Both labelling techniques were tested with two GO annotation terms, GO:0030198 (*Biol. process, extracellular matrix organization*), containing 273 genes and GO:0008284 (*Biol. process, up regulation of cell proliferation*), containing 158 genes. These two GO terms were selected, because the labels do not change significantly for GO:0030198, but do so for GO:0008284, especially for the stem cell (H1-hESC). The differences in labels are as follows.

	Comparison to CD20($t = 0.7$)	
Cell Type	GO:0008284 (273 Bags)	GO:0030198 (158 Bags)
GM12878	132(-), 141(+)	77(-), 81(+)
K562	121(-), 152(+)	78(-), 80(+)
H1-hESC	84(-), 189(+)	19(-), 139(+)
	Arbitrary Threshold ($t' = 0.007$)	
Cell Type	GO:0008284 (273 Bags)	GO:0030198 (158 Bags)
GM12878	114(-), 159(+)	81(-), 77(+)
K562	110(-), 163(+)	77(-), 81(+)
H1-hESC	56(-), 217(+)	20(-), 138(+)

Table 5.4: Output labels of the two different discretization schemes.

The results of the two 5-fold cross validation experiments are given in Tables 5.5 and 5.6. For each GO term tested, the idea is to perform 5-fold cross validation on the dataset of each individual cell type and then vertically concatenate the datasets. In this way, the labels will be mixed on purpose, to reduce classification performance. And this is indeed what can be observed on all 4 cases. An initial observation is that the comparison scheme does not work as well as the arbitrary one across all cell types. The performance is marginally better or even worse than random. What is astonishing is that, for GO:0030198 that labels are almost the same between the two schemes, the arbitrary threshold t' significantly outperforms the comparing one. The next step is to change these thresholds slightly, to evaluate robustness.

GO:0008284		
Comparison to CD20($t = 0.7$**)**		
GM12878	AUC	error
Citation MIL	51.3 (8.3)	48.3 (7.8)
Simple MIL (Bayes-Normal-1, q=0.01)	55.1 (6.6)	43.2 (3.8)
MILES (r=5.0)	51.0 (5.7)	48.3 (0.4)
K562	AUC	error
Citation MIL	56.7 (3.6)	43.6 (2.1)
Simple MIL (Bayes-Normal-1, q=0.01)	63.8 (7.5)	42.9 (6.1)
MILES (r=5.0)	52.6 (7.9)	44.3 (0.4)
H1-hESC	AUC	error
Citation MIL	61.4 (7.0)	31.9 (3.6)
Simple MIL (Bayes-Normal-1, q=0.01)	60.0 (8.9)	30.1 (1.9)
MILES (r=5.0)	52.0 (4.5)	30.8 (0.3)
All 3 Cell Types (43(-), 230(+))	AUC	error
Citation MIL	47.4 (11.0)	22.0 (4.4)
Simple MIL (Bayes-Normal-1, q=0.01)	44.3 (9.4)	15.7 (0.8)
MILES (r=5.0)	49.9 (6.2)	15.7 (0.8)
Arbitrary Threshold ($t' = 0.007$**)**		
GM12878	AUC	error
Citation MIL	80.3 (6.2)	23.5 (2.6)
Simple MIL (Bayes-Normal-1, q=0.01)	89.1 (4.0)	18.0 (3.1)
MILES (r=5.0)	88.2 (4.5)	21.9 (5.1)
K562	AUC	error
Citation MIL	81.9 (8.3)	21.6 (8.1)
Simple MIL (Bayes-Normal-1, q=0.01)	90.5 (5.1)	17.2 (3.8)
MILES (r=5.0)	88.5 (4.1)	20.9 (4.2)
H1-hESC	AUC	error
Citation MIL	72.7 (11.6)	23.0 (6.0)
Simple MIL (Bayes-Normal-1, q=0.01)	86.7 (5.4)	20.5 (3.9)
MILES (r=5.0)	70.4 (16.4)	20.5 (0.5)
All 3 Cell Types (39(-), 234(+))	AUC	error
Citation MIL	73.0 (14.4)	14.7 (4.1)
Simple MIL (Bayes-Normal-1, q=0.01)	88.9 (6.6)	13.6 (1.7)
MILES (r=5.0)	66.0 (16.6)	14.6 (0.6)

Table 5.5: Classification outcomes for GO:0008284.

GO:0030198		
Comparison to CD20(t = 0.7)		
GM12878	AUC	error
Citation MIL	55.3 (4.4)	46.3 (7.4)
Simple MIL (Bayes-Normal-1, q=0.01)	45.6 (11.5)	49.5 (10.0)
MILES (r=5.0)	47.1 (9.4)	48.7 (0.7)
K562	AUC	error
Citation MIL	59.3 (10.7)	39.9 (9.3)
Simple MIL (Bayes-Normal-1, q=0.01)	62.9 (13.0)	36.8 (8.8)
MILES (r=5.0)	51.2 (13.2)	49.4 (0.9)
H1-hESC	AUC	error
Citation MIL	52.0 (10.3)	17.1 (3.8)
Simple MIL (Bayes-Normal-1, q=0.01)	56.2 (11.2)	12.0 (1.1)
MILES (r=5.0)	55.3 (10.9)	12.0 (1.1)
All 3 Cell Types (15(-), 143(+))	AUC	error
Citation MIL	46.0 (11.1)	15.2 (2.9)
Simple MIL (Bayes-Normal-1, q=0.01)	45.0 (20.6)	9.5 (0.2)
MILES (r=5.0)	47.8 (4.9)	9.5 (0.2)
Arbitrary Threshold (t′ = 0.007)		
GM12878	AUC	error
Citation MIL	71.4 (9.5)	29.4 (8.1)
Simple MIL (Bayes-Normal-1, q=0.01)	77.7 (7.3)	27.7 (6.6)
MILES (r=5.0)	72.6 (9.1)	22.7 (6.0)
K562	AUC	error
Citation MIL	77.2 (6.8)	24.7 (4.8)
Simple MIL (Bayes-Normal-1, q=0.01)	87.5 (8.5)	18.4 (7.4)
MILES (r=5.0)	83.5 (8.8)	22.7 (6.9)
H1-hESC	AUC	error
Citation MIL	62.6 (9.7)	14.6 (3.6)
Simple MIL (Bayes-Normal-1, q=0.01)	75.1 (12.8)	15.2 (3.3)
MILES (r=5.0)	42.8 (13.0)	12.7 (0.2)
All 3 Cell Types (15(-), 143(+))	AUC	error
Citation MIL	57.6 (2.9)	10.8 (2.8)
Simple MIL (Bayes-Normal-1, q=0.01)	68.9 (12.6)	12.6 (3.8)
MILES (r=5.0)	64.8 (14.7)	9.5 (0.2)

Table 5.6: Classification Results for GO:0030198.

5.2.4 Perturbing the thresholds

The decision for setting the new thresholds is twofold. On one hand, the intuition from biology is that most of the genes should be expressed at some point during the life cycle of a cell, resulting in a larger positive class. On the other hand, with the previous settings, the labels differed greatly between the two schemes. So, the threshold t was decreased to 0.4, to allow more genes to be labelled positive, while the threshold t' was increased to 0.03, to decrease the number of positive genes. The results can be seen in the Table 5.7. The numbers are quite similar across the same cell type.

	Positive Genes (out of 12570)	
Cell Type	Comparison to CD20($t = 0.4$)	Arbitrary Threshold ($t' = 0.03$)
GM12878	7818	8074
K562	8040	8096
H1-hESC	9874	9653

Table 5.7: Output labels of the two different discretization schemes with the new thresholds.

The labels for the two GO term datasets are given below. They are identical for the two methods.

	Comparison to CD20($t = 0.4$)	
Cell Type	GO:0008284 (273 Bags)	GO:0030198 (158 Bags)
GM12878	107(-), 166(+)	66(-), 92(+)
K562	103(-), 170(+)	71(-), 87(+)
H1-hESC	60(-), 213(+)	15(-), 143(+)
	Arbitrary Threshold ($t' = 0.03$)	
Cell Type	GO:0008284 (273 Bags)	GO:0030198 (158 Bags)
GM12878	107(-), 166(+)	66(-), 92(+)
K562	103(-), 170(+)	71(-), 87(+)
H1-hESC	60(-), 213(+)	15(-), 143(+)

Table 5.8: Output labels of the two different discretization schemes for the two GO terms.

The classification results are given in Tables 5.9 and 5.10. The results are identical across the two methods for each GO term. Overall, they are very poor, but at least they demonstrate that the high performance of the initial setting of the arbitrary threshold was a coincidence. Furthermore, the new results show that, as more bags become positive, there is a slight improvement in performance. This fact does not

explain the vast difference in performance between the two methods in the previous section for GO:0030198, where the bag labels are almost identical. The bags that change labels may be the most significant for classification.

GO:0008284		
Comparison to CD20($t = 0.4$**)**		
GM12878	AUC	error
Citation MIL	53.6 (8.1)	43.9 (5.8)
Simple MIL (Bayes-Normal-1, q=0.01)	61.2 (3.2)	40.3 (2.3)
MILES (r=5.0)	57.2 (9.4)	39.2 (0.5)
K562	AUC	error
Citation MIL	61.7 (5.2)	36.6 (2.1)
Simple MIL (Bayes-Normal-1, q=0.01)	68.9 (9.9)	35.5 (10.5)
MILES (r=5.0)	53.8 (6.1)	37.7 (0.6)
H1-hESC	AUC	error
Citation MIL	65.5 (6.5)	24.5 (2.7)
Simple MIL (Bayes-Normal-1, q=0.01)	68.7 (4.9)	26.0 (4.1)
MILES (r=5.0)	55.4 (5.7)	22.0 (0.2)
Arbitrary Threshold ($t' = 0.03$**)**		
GM12878	AUC	error
Citation MIL	53.6 (8.1)	43.9 (5.8)
Simple MIL (Bayes-Normal-1, q=0.01)	61.2 (3.2)	40.3 (2.3)
MILES (r=5.0)	57.2 (9.4)	39.2 (0.5)
K562	AUC	error
Citation MIL	61.7 (5.2)	36.6 (2.1)
Simple MIL (Bayes-Normal-1, q=0.01)	68.9 (9.9)	35.5 (10.5)
MILES (r=5.0)	53.8 (6.1)	37.7 (0.6)
H1-hESC	AUC	error
Citation MIL	65.5 (6.5)	24.5 (2.7)
Simple MIL (Bayes-Normal-1, q=0.01)	68.7 (4.9)	26.0 (4.1)
MILES (r=5.0)	55.4 (5.7)	22.0 (0.2)

Table 5.9: Classification outcomes for GO:0008284 with the new thresholds.

GO:0030198		
Comparison to CD20($t = 0.4$**)**		
GM12878	AUC	error
Citation MIL	61.2 (1.7)	37.3 (5.6)
Simple MIL (Bayes-Normal-1, q=0.01)	50.1 (4.2)	46.2 (3.5)
MILES (r=5.0)	46.3 (2.4)	41.8 (0.7)
K562	AUC	error
Citation MIL	58.9 (12.1)	39.2 (8.0)
Simple MIL (Bayes-Normal-1, q=0.01)	66.2 (7.9)	35.5 (4.6)
MILES (r=5.0)	50.2 (12.7)	44.9 (0.7)
H1-hESC	AUC	error
Citation MIL	52.6 (9.1)	12.7 (3.3)
Simple MIL (Bayes-Normal-1, q=0.01)	66.5 (13.9)	10.8 (1.9)
MILES (r=5.0)	48.7 (1.9)	9.5 (0.2)
Arbitrary Threshold ($t' = 0.03$**)**		
GM12878	AUC	error
Citation MIL	61.2 (1.7)	37.3 (5.6)
Simple MIL (Bayes-Normal-1, q=0.01)	50.1 (4.2)	46.2 (3.5)
MILES (r=5.0)	46.3 (2.4)	41.8 (0.7)
K562	AUC	error
Citation MIL	58.9 (12.1)	39.2 (8.0)
Simple MIL (Bayes-Normal-1, q=0.01)	66.2 (7.9)	35.5 (4.6)
MILES (r=5.0)	50.2 (12.7)	44.9 (0.7)
H1-hESC	AUC	error
Citation MIL	52.6 (9.1)	12.7 (3.3)
Simple MIL (Bayes-Normal-1, q=0.01)	66.5 (13.9)	10.8 (1.9)
MILES (r=5.0)	48.7 (1.9)	9.5 (0.2)

Table 5.10: Classification Results for GO:0030198 with the new thresholds.

5.2.5 Investigating Two Cell Type Classification

To further investigate the previous results, classification of a pattern was attempted with all 4 thresholds. The pattern under consideration was a gene not being expressed in GM12878 (0 label) but being expressed in K562 (1 label). This was deemed interesting as it would help determine differences between the healthy and cancerous blood cells.

For the two GO terms, the gene labels for this pattern across two cell types can be seen in Table 5.11.

	Arbitrary Thresholds		Comparison to CD20	
GO Term	0.007	0.03	0.4	0.7
0008284 (273 genes)	242(-), 31(+)	245(-), 28(+)	238(-), 35(+)	227(-), 46(+)
0030198 (158 genes)	135(-), 23(+)	137(-), 21(+)	133(-), 25(+)	133(-), 25(+)

Table 5.11: Output labels of all 4 thresholds for the pattern "G0K1" (GM12878 = 0, K562 = 1).

For GO term GO:0008284 the results are as follows:

GO:0008284		
Comparison to CD20($t = 0.7$)		
GM12878 = 0, K562 = 1	AUC	error
Citation MIL	52.8 (5.3)	29.3 (2.7)
Simple MIL (Bayes-Normal-1, q=0.01)	59.9 (13.7)	28.9 (7.3)
MILES (r=5.0)	51.2 (2.2)	16.8 (0.6)
Arbitrary Threshold ($t' = 0.007$)		
GM12878 = 0, K562 = 1	AUC	error
Citation MIL	64.7 (10.0)	16.1 (4.5)
Simple MIL (Bayes-Normal-1, q=0.01)	76.8 (10.8)	19.8 (5.4)
MILES (r=5.0)	49.8 (0.5)	11.3 (0.6)
GO:0008284		
Comparison to CD20($t = 0.4$)		
GM12878 = 0, K562 = 1	AUC	error
Citation MIL	63.7 (8.4)	18.3 (4.2)
Simple MIL (Bayes-Normal-1, q=0.01)	64.8 (12.5)	19.1 (5.2)
MILES (r=5.0)	55.3 (7.6)	12.8 (0.1)
Arbitrary Threshold ($t' = 0.03$)		
GM12878 = 0, K562 = 1	AUC	error
Citation MIL	52.6 (10.2)	17.2 (3.5)
Simple MIL (Bayes-Normal-1, q=0.01)	64.3 (4.5)	18.3 (3.4)
MILES (r=5.0)	49.4 (2.6)	10.2 (0.9)

Table 5.12: Classification outcomes for GO:0008284 for a specific pattern and both sets of thresholds.

As can be seen in Table 5.12, the classification performance for the comparing strategy (thresholds $t = 0.7$ and $t = 0.4$) is overall quite poor. Despite this, it is consistent with the performance of the same strategy for single cell types and the same GO term (0008284), as can be seen in top parts of Tables 5.5 and 5.9. The same does hold for the arbitrary threshold. While the lower threshold, $t' = 0.007$, performs really well on single cell types (Table 5.5), the performance drops significantly for pattern classifica-

tion. With the higher threshold, $t' = 0.03$, performance is similar (Table 5.9), but does not behave as well as the comparison scheme.

For GO:0030198 the corresponding results tell a similar tale.

GO:0030198		
Comparison to CD20($t = 0.7$)		
GM12878 = 0, K562 = 1	AUC	error
Citation MIL	55.2 (5.5)	23.4 (4.9)
Simple MIL (Bayes-Normal-1, q=0.01)	65.2 (15.6)	21.5 (9.3)
MILES (r=5.0)	51.2 (7.6)	15.8 (0.3)
Arbitrary Threshold ($t' = 0.007$)		
GM12878 = 0, K562 = 1	AUC	error
Citation MIL	66.6 (10.8)	17.1 (6.1)
Simple MIL (Bayes-Normal-1, q=0.01)	71.8 (12.3)	18.3 (4.0)
MILES (r=5.0)	52.9 (18.4)	14.5 (1.5)
GO:0030198		
Comparison to CD20($t = 0.4$)		
GM12878 = 0, K562 = 1	AUC	error
Citation MIL	56.7 (13.6)	25.9 (4.1)
Simple MIL (Bayes-Normal-1, q=0.01)	63.0 (14.5)	22.7 (4.4)
MILES (r=5.0)	52.6 (3.7)	15.8 (0.3)
Arbitrary Threshold ($t' = 0.03$)		
GM12878 = 0, K562 = 1	AUC	error
Citation MIL	54.4 (10.3)	19.5 (6.6)
Simple MIL (Bayes-Normal-1, q=0.01)	67.7 (12.6)	15.7 (6.7)
MILES (r=5.0)	57.0 (11.7)	13.3 (1.1)

Table 5.13: Classification outcomes for GO:0030198 for a specific pattern and all thresholds.

The performance is consistent between single cell type and pattern classification (compare $t = 0.7$ of Table 5.13 to top of Table 5.6 and $t = 0.4$ of Table 5.13 to top of Table 5.10). For the arbitrary thresholds, $t' = 0.007$ and $t' = 0.03$, the performance is not the same across the two scenarios. For the lower threshold,$t' = 0.007$, the performance decreases (compare to middle of Table 5.10), while for the higher threshold, $t' = 0.03$, it is similar (compare to middle of Table 5.10).

From this analysis, it can be seen that there is an increasing trend in the single cell type, as well as the pattern, classification performance for the comparison to another cell type scheme, as more genes migrate to the positive class. The same cannot be said for the arbitrary thresholds, as the performance either deteriorates severely ($t' = 0.007$), or stays similar ($t' = 0.03$) between the single cell and pattern classifications. This makes the comparison scheme more robust and promising for pattern search.

6 December 2013

6.1 Discretization Strategies

6.1.1 0-1-0 Scheme

Since the first comparative scheme did not yield very good results, another one was devised. If a gene of interest is expressed *at the same level* as in the reference cell type, then assign a label 1, otherwise assign a label 0, if it is significantly over- or under-expressed. The scheme can be visualized in the following Figure.

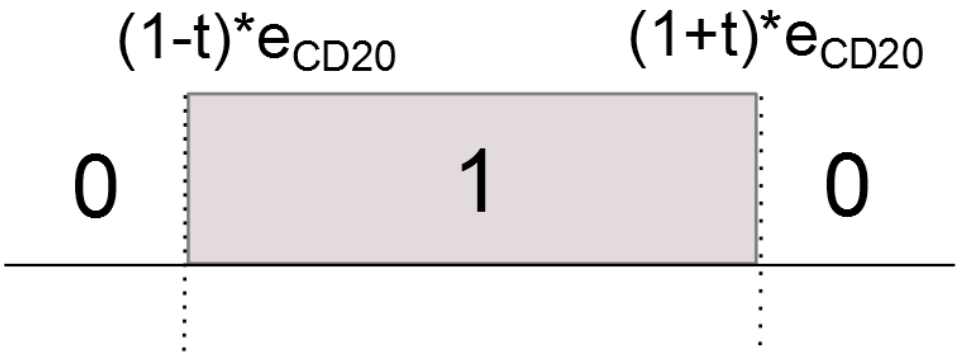


Figure 6.1: The new comparison scheme with reference cell type CD20.

	Comparison to CD20($t = 1 - 0.3$ **and** $t = 1 + 0.3$**)**	
Cell Type	GO:0008284 (273 Bags)	GO:0030198 (158 Bags)
GM12878	19(-), 254(+)	11(-), 147(+)
K562	21(-), 252(+)	18(-), 140(+)
H1-hESC	5(-), 268(+)	3(-), 155(+)

Table 6.1: Output labels of the 0-1-0 discretization scheme for the two GO terms.

Comparison to CD20($t = 1 - 0.3$ and $t = 1 + 0.3$)

GO:0008284		
GM12878	AUC	error
Citation MIL	48.9 (9.9)	9.2 (1.3)
Simple MIL (Bayes-Normal-1, q=0.01)	86.0 (8.6)	12.1 (4.8)
MILES (r=5.0)	69.4 (11.7)	7.0 (0.7)
K562	AUC	error
Citation MIL	70.1 (5.9)	11.4 (1.7)
Simple MIL (Bayes-Normal-1, q=0.01)	83.9 (11.3)	9.5 (2.4)
MILES (r=5.0)	40.2 (8.6)	7.7 (0.7)
H1hESC	AUC	error
Citation MIL	66.8 (27.0)	2.6 (2.1)
Simple MIL (Bayes-Normal-1, q=0.01)	88.4 (7.5)	2.2 (0.8)
MILES (r=5.0)	76.4 (16.2)	1.8 (0.0)
GO:0030198		
GM12878	AUC	error
Citation MIL	62.1 (14.9)	9.5 (2.3)
Simple MIL (Bayes-Normal-1, q=0.01)	77.6 (10.0)	10.8 (3.6)
MILES (r=5.0)	49.3 (13.0)	6.9 (1.2)
K562	AUC	error
Citation MIL	61.1 (14.7)	15.2 (5.2)
Simple MIL (Bayes-Normal-1, q=0.01)	83.4 (10.4)	15.7 (7.2)
MILES (r=5.0)	51.0 (2.2)	11.4 (1.5)
H1hESC	AUC	error
Citation MIL	-	-
Simple MIL (Bayes-Normal-1, q=0.01)	-	-
MILES (r=5.0)	-	-

Table 6.2: Classification outcomes for GO:0008284 with thresholds 0.7-1.3. For cell type H1hESC, only 3 bags were negative and CV could not be performed.

6.1.2 Absolute Zero

Here the discretization is the same as using an absolute threshold, with the only difference that the threshold is zero. If a gene has 0 RPKM expression value, give 0 label, 1 otherwise.

Absolute Zero		
Cell Type	GO:0008284 (273 Bags)	GO:0030198 (158 Bags)
GM12878	32(-), 241(+)	17(-), 141(+)
K562	43(-), 230(+)	27(-), 131(+)
H1-hESC	13(-), 260(+)	3(-), 155(+)

Table 6.3: Output labels of the zero discretization scheme for the two GO terms.

Absolute zero		
GO:0008284		
GM12878	AUC	error
Citation MIL	63.8 (10.5)	14.6 (2.2)
Simple MIL (Bayes-Normal-1, q=0.01)	84.0 (6.4)	16.8 (6.8)
MILES (r=5.0)	70.7 (15.3)	11.7 (0.8)
K562	AUC	error
Citation MIL	70.8 (7.4)	19.8 (5.8)
Simple MIL (Bayes-Normal-1, q=0.01)	83.2 (10.3)	18.3 (8.6)
MILES (r=5.0)	50.0 (0.7)	15.7 (0.8)
H1hESC	AUC	error
Citation MIL	69.0 (6.1)	5.5 (1.3)
Simple MIL (Bayes-Normal-1, q=0.01)	77.4 (8.8)	6.2 (2.0)
MILES (r=5.0)	62.8 (20.9)	4.8 (1.0)
GO:0030198		
GM12878	AUC	error
Citation MIL	59.1 (16.9)	17.7 (3.2)
Simple MIL (Bayes-Normal-1, q=0.01)	73.8 (16.6)	11.3 (3.9)
MILES (r=5.0)	54.8 (10.0)	10.7 (1.4)
K562	AUC	error
Citation MIL	52.5 (12.3)	24.0 (4.7)
Simple MIL (Bayes-Normal-1, q=0.01)	80.6 (11.0)	23.3 (8.8)
MILES (r=5.0)	59.7 (10.9)	17.1 (1.3)
H1hESC	AUC	error
Citation MIL	-	-
Simple MIL (Bayes-Normal-1, q=0.01)	-	-
MILES (r=5.0)	-	-

Table 6.4: Classification outcomes for both GO terms with the absolute zero thresholds. Because of too few negative bags, CV could not be done for H1hESC.

6.1.3 Tables for 2 Cell Types

GO:0008284		
Comparison to CD20($t = 0.7$**)**		
GM12878 = 0, K562 = 1	AUC	error
Citation MIL	52.8 (5.3)	29.3 (2.7)
Simple MIL (Bayes-Normal-1, q=0.01)	59.9 (13.7)	28.9 (7.3)
MILES (r=5.0)	51.2 (2.2)	16.8 (0.6)
Comparison to CD20($t = 0.4$**)**		
GM12878 = 0, K562 = 1	AUC	error
Citation MIL	63.7 (8.4)	18.3 (4.2)
Simple MIL (Bayes-Normal-1, q=0.01)	64.8 (12.5)	19.1 (5.2)
MILES (r=5.0)	55.3 (7.6)	12.8 (0.1)
Comparison to CD20($t = 1 - 0.3$ **and** $t = 1 + 0.3$**)**		
GM12878 = 0, K562 = 1	AUC	error
Citation MIL	52.1 (12.1)	4.0 (1.5)
Simple MIL (Bayes-Normal-1, q=0.01)	77.8 (10.1)	15.0 (5.9)
MILES (r=5.0)	50.2 (0.4)	2.6 (1.0)
Arbitrary Threshold ($t' = 0.007$**)**		
GM12878 = 0, K562 = 1	AUC	error
Citation MIL	64.7 (10.0)	16.1 (4.5)
Simple MIL (Bayes-Normal-1, q=0.01)	76.8 (10.8)	19.8 (5.4)
MILES (r=5.0)	49.8 (0.5)	11.3 (0.6)
Arbitrary Threshold ($t' = 0.03$**)**		
GM12878 = 0, K562 = 1	AUC	error
Citation MIL	52.6 (10.2)	17.2 (3.5)
Simple MIL (Bayes-Normal-1, q=0.01)	64.3 (4.5)	18.3 (3.4)
MILES (r=5.0)	49.4 (2.6)	10.2 (0.9)
Absolute zero		
GM12878 = 0, K562 = 1	AUC	error
Citation MIL	55.4 (18.4)	9.5 (2.7)
Simple MIL (Bayes-Normal-1, q=0.01)	78.4 (6.2)	18.0 (4.2)
MILES (r=5.0)	50.0 (0.0)	5.1 (0.8)

Table 6.5: Classification outcomes for GO:0008284 for a specific pattern and all thresholds. Repeating results from Table 5.12.

GO:0030198		
Comparison to CD20($t = 0.7$**)**		
GM12878 = 0, K562 = 1	AUC	error
Citation MIL	55.2 (5.5)	23.4 (4.9)
Simple MIL (Bayes-Normal-1, q=0.01)	65.2 (15.6)	21.5 (9.3)
MILES (r=5.0)	51.2 (7.6)	15.8 (0.3)
Comparison to CD20($t = 0.4$**)**		
GM12878 = 0, K562 = 1	AUC	error
Citation MIL	56.7 (13.6)	25.9 (4.1)
Simple MIL (Bayes-Normal-1, q=0.01)	63.0 (14.5)	22.7 (4.4)
MILES (r=5.0)	52.6 (3.7)	15.8 (0.3)
Comparison to CD20($t = 1 - 0.3$ **and** $t = 1 + 0.3$**)**		
GM12878 = 0, K562 = 1	AUC	error
Citation MIL	76.1 (27.6)	6.3 (2.2)
Simple MIL (Bayes-Normal-1, q=0.01)	54.4 (16.4)	7.0 (1.5)
MILES (r=5.0)	50.0 (0.0)	3.2 (0.1)
Arbitrary Threshold ($t' = 0.007$**)**		
GM12878 = 0, K562 = 1	AUC	error
Citation MIL	66.6 (10.8)	17.1 (6.1)
Simple MIL (Bayes-Normal-1, q=0.01)	71.8 (12.3)	18.3 (4.0)
MILES (r=5.0)	52.9 (18.4)	14.5 (1.5)
Arbitrary Threshold ($t' = 0.03$**)**		
GM12878 = 0, K562 = 1	AUC	error
Citation MIL	54.4 (10.3)	19.5 (6.6)
Simple MIL (Bayes-Normal-1, q=0.01)	67.7 (12.6)	15.7 (6.7)
MILES (r=5.0)	57.0 (11.7)	13.3 (1.1)
Absolute zero		
GM12878 = 0, K562 = 1	AUC	error
Citation MIL	57.4 (18.1)	13.3 (5.0)
Simple MIL (Bayes-Normal-1, q=0.01)	66.8 (19.3)	17.0 (6.1)
MILES (r=5.0)	54.3 (6.0)	6.9 (1.2)

Table 6.6: Classification outcomes for GO:0030198 for a specific pattern and all thresholds. Repeating results from Table 5.13.

6.1.4 Looking at different thresholds.

To see how performance varies when the label discretization threshold is perturbed the test dataset that performed best was selected. That was for GO term 0008284 and for cell type K562 (Table 5.5). This dataset was relabelled using the following thresholds: $0, 0.003, 0.004, 0.005, 0.006, 0.007, 0.008, 0.009, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08$. Then the 5-fold cross validation experiment was repeated for the three classifiers and

the results can be in Figure 6.2.

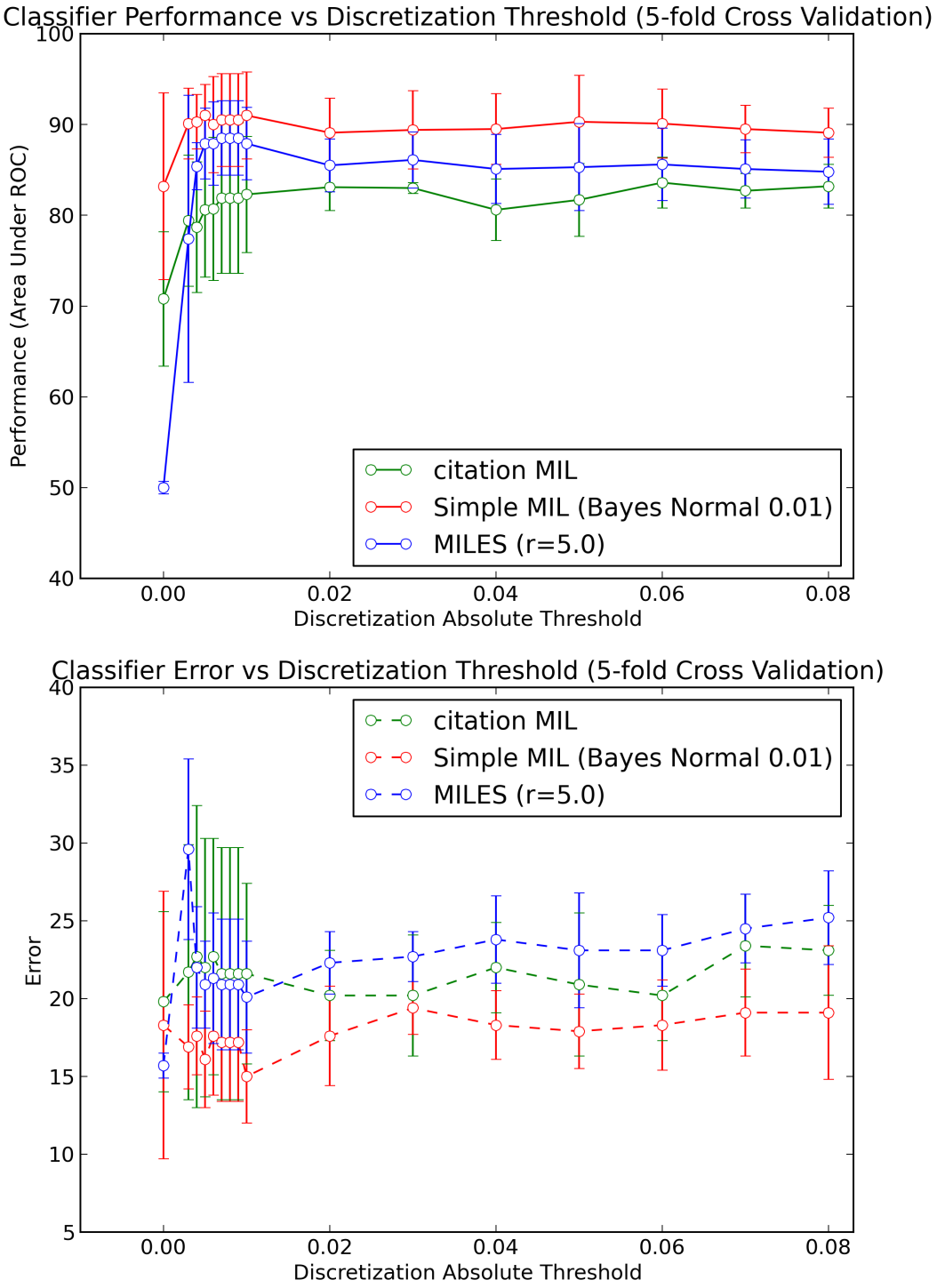


Figure 6.2: The new comparison scheme with reference cell type CD20.

ATTENTION: For this experiment, I made the test data files again. The results of

0.007 are identical between this experiment and the original one (Table 5.5). For absolute threshold 0.03 though, the results are NOT the same (Table 5.9 and 83.0 (0.6), 89.4 (4.3), 86.1 (3.1)). The labels differ (originally 103(-), 170(+), but now 129(-), 144(+)). I am looking into my code.

6.2 Updated Dataset

As of this point, the promoter regions for all genes were considered from the transcription start site plus or minus a thousand base-pairs according to the strand. But this is not correct, as the bases on the negative strand are still increasing from the 5′ end to the 3′ end, but are read in reverse. So, the end of a gene on the negative strand is the actual transcription start site. With this consideration in mind, the association of TFs to genes was run again using a file generated from BioMart that contains only the beginning and end of genes without considering transcripts. This resulted in a new dataset containing 18579 genes and 192K instances.

For this dataset, the 1-0-1 labelling scheme (significantly different from reference) was used, with NHEK (skin cells) from Tier 3 used as reference, as CD20 did not contain expression data for all the genes in the updated dataset.

There are 82 distinct TFs associated to the genes of the dataset. For those, 82 new binary features were constructed and added to each instance in the dataset. For the presence of a certain TF, the corresponding feature gets value 1, while the rest remain 0. At this point, the features of open chromatin and histone modifications were modified. For the open chromatin, an extra column was added with the maximum distance of the TF from the peak. For histone modifications, the maximum was used instead of the average.

With these features added, resulting to 100 features, the linear classifier used in Simple MIL could not work. Some tweaks were needed for the current Matlab installation.

6.2.1 Matlab Tweaks

The linear classifier used with Simple MIL was crushing with the addition of the extra features. For this reason, the logistic linear classifier was used instead. To do so, PR Tools was updated to the current version 5. The `minFunc` package by Mark Schmidt was added. The SLEP (Sparse Learning with Efficient Projections)[1] package [14] was also added to use with the implementation of a sparse logistic classifier.

[1] http://www.public.asu.edu/ jye02/Software/SLEP/

7 January 2014

This month was solely devoted to writing the article form of the thesis. The switch from focus on the details to the more high-level and abstract thinking was a real challenge. The most important figures describing the overview of the implementation were created and are given below.

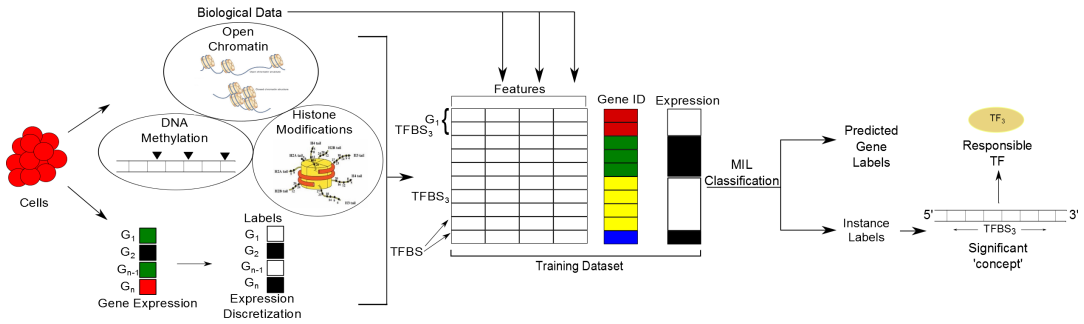


Figure 7.1: The overview for significant TF discovery in a single cell type.

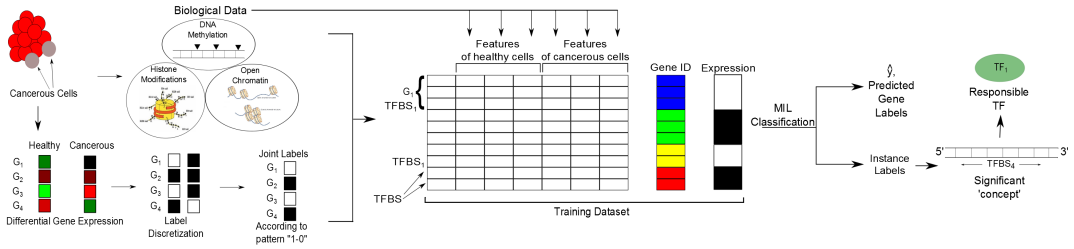


Figure 7.2: The overview for significant TF discovery in multiple cell types. In this case, the search is done for a pattern, e.g. "1-0", i.e. a gene is expressed in a healthy cell, but is not expressed in a cancerous one.

8 February 2014

8.1 Updated Dataset

8.1.1 Correct TFBSs association to genes

As was mentioned before, the counting of bps on the genome is done from left to right. This is the same on the positive strand, but on the negative strand, where the genome is read in reverse, this means that the start and end of a characteristic are, in reality, reversed. As an example, this is illustrated for the transcription start site (TSS) in Fig. 8.1.

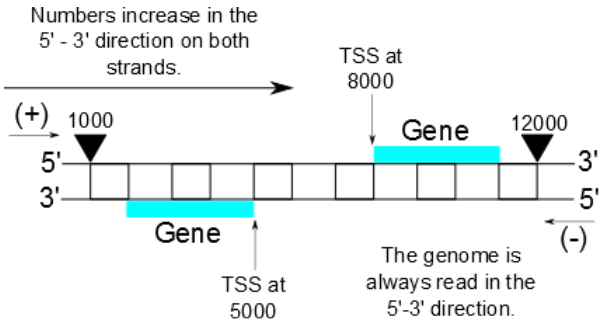


Figure 8.1: A toy example of the TSS for two genes, one on each strand. For the gene on the negative strand, the TSS is the given end of the gene.

The assumption made in this study is that, if the start of a TF peak falls within 1000 bps of the TSS, then this TFBS is associated to a particular gene. This means that, on the negative strand, the position given as the end of the peak is the actual start. With this consideration, an updated version of the dataset was generated.

8.1.2 Gene Expression Data

As a new dataset was created, TFBSs were associated to 15782 protein-coding genes. Since the start of this study, the RNA-seq data used for gene expression were from the Cold Spring Harbor Lab. But, this database turned out to be incomplete. When trying to associate a gene expression value, 603 genes were omitted. For this reason, an alternate to this was used, namely the database from Caltech. This contains data for fewer cell types, but for larger gene sets. The only difference is that the values

are not normalized over all biological replicates, so the mean was used. Only 5 genes were omitted.

The labelling scheme was 0-1-0, significantly similar to the reference, using gene expression from NHEK (a skin cell) as reference. The resulting labels can be seen in Table 8.1.

Positive Genes (out of 15008)	
Cell Type	
GM12878	8365
K562	8223
H1-hESC	8699

Table 8.1: Output labels of the 0-1-0 comparison discretization scheme.

8.1.3 Classification Results

As before, 82 binary features, representing the presence of each of the unique TFs, were added to the dataset. When scaling the dataset to unit variance, for the Citation MIL and MILES classifiers, these features remain unscaled. For GO term 0016055, representing the WNT pathway, 125 genes of the dataset were identified with 2004 instances in total. The labels for this subset of genes can be seen in Table 8.2.

Comparison to NHEK($t = 1 - 0.6$ **and** $t = 1 + 0.6$**)**	
Cell Type	GO:0016055 (125 Bags)
GM12878	50(-), 75(+)
K562	53(-), 72(+)
H1-hESC	44(-), 81(+)
G0K1	117(-), 8(+)

Table 8.2: Output labels of the 0-1-0 comparison discretization scheme for GO term 0016055.

The classification results using partial scaling, and a nearest neighbour distance for the radial kernel of MILES can be seen in Table 8.3.

Comparison to NHEK($t = 1 - 0.6$ and $t = 1 + 0.6$)		
GO:0016055		
GM12878	AUC	error
Citation MIL	65.9 (16.6)	29.6 (7.8)
Simple MIL (Logistic2, q=0.01)	77.5 (10.4)	25.6 (8.8)
MILES (r=NaN)	69.1 (18.4)	37.6 (17.6)
K562	AUC	error
Citation MIL	52.0 (12.2)	41.4 (12.6)
Simple MIL (Logistic2, q=0.01)	67.3 (8.0)	31.9 (6.2)
MILES (r=NaN)	60.6 (9.0)	44.8 (7.1)
H1hESC	AUC	error
Citation MIL	63.9 (10.4)	34.5 (6.3)
Simple MIL (Logistic2, q=0.01)	68.1 (9.3)	31.1 (3.6)
MILES (r=NaN)	63.7 (7.1)	36.1 (7.8)
GM12878 = 0, K562 = 1	AUC	error
Citation MIL	70.1 (25.7)	8.0 (2.8)
Simple MIL (Logistic2, q=0.01)	54.3 (34.6)	13.6 (3.5)
MILES (r=NaN)	57.8 (31.1)	6.3 (2.0)

Table 8.3: Classification outcomes for GO term 0016055 with the 0-1-0, NHEK comparing, 0.6 threshold.

Naming Decisions

While writing the paper, the decision was made to dub the absolute arbitrary threshold as θ and the strategy y_θ and the 0-1-0 strategy as y_{010}. Henceforth, all references to the two strategies will use these names.

8.2 Best Results

8.2.1 RPKM value distribution

With the new values from Caltech, the distribution of expression values for all genes from the 4 cell types (GM12878, K562, H1-hESC and NHEK) was rebuilt.

Based on this distribution, the thresholds used for the y_θ strategy were $0, 7.5, 12.5, 17.5, 20, 30, 40, 50, 60, 70, 80$. For the y_{010} strategy, the threshold, t, was varied between $0.3 - 0.9$ in increments of 0.1.

From this point on, all the work that was done was put directly in the paper and the writing of the log was abandoned. If not acceptable, it will be completed with the major experiments done in more detail.

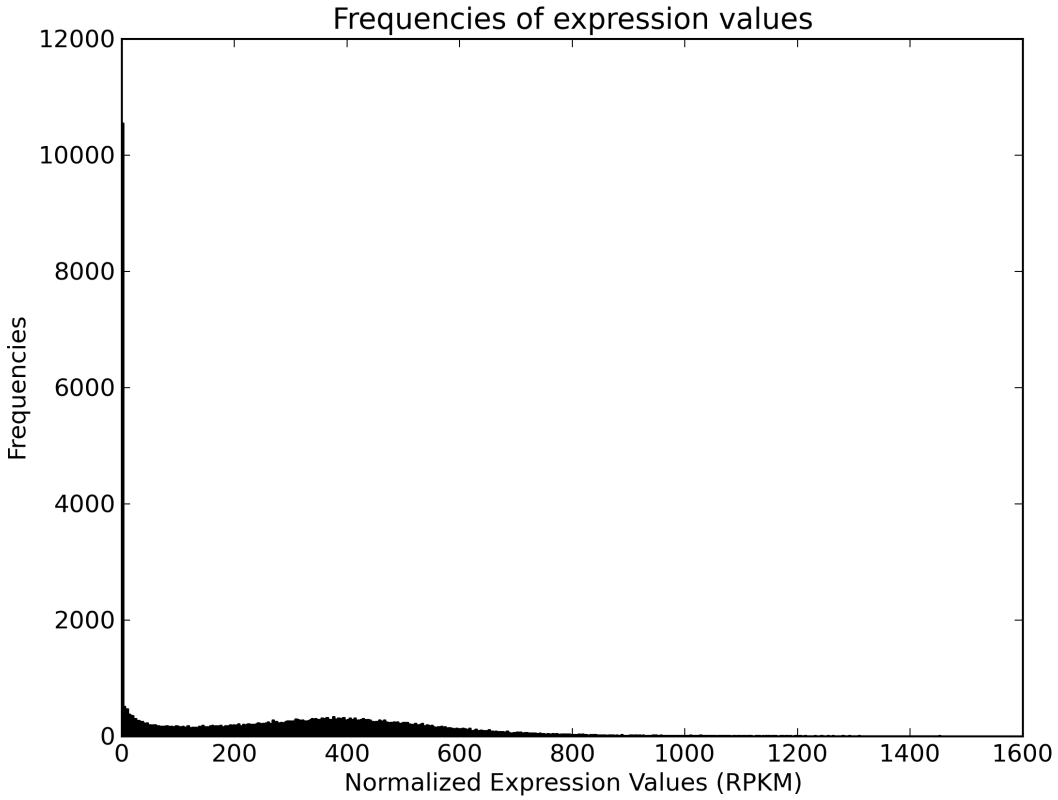


Figure 8.2: Distribution of normalized expression values, in RPKM, for all 4 cell types. Most of the genes are lowly expressed.

References

1. Wikipedia. Transcription Factor – Wikipedia, The Free Encyclopedia. 2013.
2. T Dietterich. Solving the multiple instance problem with axis-parallel rectangles. In: *Artificial Intelligence* 89.1-2 (1997), pp. 31–71.
3. Yixin Chen, Jinbo Bi, and James Z. Wang. MILES: Multiple-Instance Learning via Embedded Instance Selection. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.12 (2006), pp. 1931–1947.
4. Wikipedia. ChIP-sequencing – Wikipedia, The Free Encyclopedia. 2013.
5. Peter J Park. ChIP-seq: advantages and challenges of a maturing technology. In: *Nature reviews. Genetics* 10.10 (2009), pp. 669–680.
6. Wikipedia. Histone – Wikipedia, The Free Encyclopedia. 2013.
7. Susanne Marije Kooistra and Kristian Helin. Molecular mechanisms and potential functions of histone demethylases. In: *Nature Reviews Molecular Cell Biology* (2012).
8. Qunhua Li et al. Measuring reproducibility of high-throughput experiments. In: *The Annals of Applied Statistics* 5.3 (2011), pp. 1752–1779.
9. Juan M Vaquerizas et al. A census of human transcription factors: function, expression and evolution. In: *Nature reviews. Genetics* 10.4 (2009), pp. 252–263.
10. Weihua Zeng and Ali Mortazavi. Technical considerations for functional sequencing assays. In: *Nature Immunology* 13.9 (2012), pp. 802–807.
11. Illumina. Estimating Sequencing Coverage. 2013.
12. Günter P. Wagner, Koryu Kin, and Vincent J. Lynch. Measurement of mRNA abundance using RNA-seq data: RPKM measure is inconsistent among samples. In: *Theory in Biosciences* 131.4 (2012), pp. 281–285.
13. D. Parkhomchuk et al. Transcriptome analysis by strand-specific sequencing of complementary DNA. In: *Nucleic Acids Research* 37.18 (2009), e123–e123.
14. J. Liu, S. Ji, and J. Ye. SLEP: Sparse Learning with Efficient Projections. Arizona State University. 2009.