Analysis and Reconstruction of the Hematopoietic Stem Cell Differentiation Tree: A Linear Programming Approach for Gene Selection

Submitted by
Mohamed A. Ghadie[1,2]

(December 2014)

Supervisor: Theodore J. Perkins[1,2,3]
Co-supervisor: Nathalie Japkowicz[2]

A thesis submitted in partial fulfilment of the requirements
for the degree of Masters in Applied Science in
Electrical and Computer Engineering

Faculty of Graduate Studies
University of Ottawa
Ottawa, Ontario, Canada

[1]Sprott Center for Stem Cell Research, Ottawa Hospital Research Institute
[2]School of Electrical Engineering and Computer Science, University of Ottawa
[3]Department of Biochemistry, Microbiology and Immunology, University of Ottawa

AKNOWLEDGEMENTS

Thank you for investing your time and attention in reading this thesis. I would also like thank my two supervisors Dr. Theodore J. Perkins and Dr. Nathalie Japkowicz for their inspirations and valuable contributions that helped me complete this project. Their constructive criticism and generous feedback were critical success factors in the production of the results presented in this thesis.

Funding: This work was supported by a grant from the Natural Sciences and Engineering Research Council (NSERC) to Theodore J. Perkins, and a Queen Elizabeth II Graduate Scholarship in Science and Technology (QEII-GSST) to Mohamed A. Ghadie.

ABSTRACT

Stem cells differentiate through an organized hierarchy of intermediate cell types to terminally differentiated cell types. This process is largely guided by master transcriptional regulators, but it also depends on the expression of many other types of genes. The discrete cell types in the differentiation hierarchy are often identified based on the expression or non-expression of certain marker genes. Historically, these have often been various cell-surface proteins, which are fairly easy to assay biochemically but are not necessarily causative of the cell type, in the sense of being master transcriptional regulators. This raises important questions about how gene expression across the whole genome controls or reflects cell state, and in particular, differentiation hierarchies. Traditional approaches to understanding gene expression patterns across multiple conditions, such as principal components analysis or K-means clustering, can group cell types based on gene expression, but they do so without knowledge of the differentiation hierarchy. Hierarchical clustering and maximization of parsimony can organize the cell types into a tree, but in general this tree is different from the differentiation hierarchy. Using hematopoietic differentiation as an example, we demonstrate how many genes other than marker genes are able to discriminate between different branches of the differentiation tree by proposing two models for detecting genes that are up-regulated or down-regulated in distinct lineages. We then propose a novel approach to solving the following problem: Given the differentiation hierarchy and gene expression data at each node, construct a weighted Euclidean distance metric such that the minimum spanning tree with respect to that metric is precisely the given differentiation hierarchy. We provide a set of linear constraints that are provably sufficient for the desired construction and a linear programming framework to identify sparse sets of weights, effectively identifying genes that are most relevant for discriminating different parts of the tree. We apply our method to microarray gene expression data describing 38 cell types in the hematopoiesis hierarchy, constructing a sparse weighted Euclidean metric that uses just 175 genes. These 175 genes are different than the marker genes that were used to identify the 38 cell types, hence offering a novel alternative way of discriminating different branches of the tree. A DAVID functional annotation analysis shows that the 175 genes reflect major processes and pathways active in different parts of the tree. However, we find that there are many alternative sets of weights that satisfy the linear constraints. Thus, in the style of random-forest training, we also construct metrics based on random subsets of the genes and compare them to the metric of 175 genes. Our results show that the 175 genes frequently appear in the random metrics, implicating their significance from an empirical point of view as well. Finally, we show how our linear programming method is able to identify columns that were selected to build minimum spanning trees on the nodes of random variable-size matrices.

Table of Contents

1 INTRODUCTION

The differentiation of stem cells from a pluripotent state towards increasingly specialized cell types has long been conceptualized as a branching process (Reya et al. 2001). At the top of the hierarchy is a cell which, under the appropriate conditions, can ultimately be transformed into many other cell types. Conversely, the bottom of the hierarchy is populated with fully-differentiated cells, which do not undergo division and which exist to perform various functions for the organism. The intermediate-level cell types have some ability to specialize further, but are constrained in the cell types that they can become. In most circumstances, transitioning "sideways" from one branch to another, or returning to higher levels of the hierarchy, does not happen. (Regeneration in Newts and related species (Maki et al. 2009), as well as methods for generating induced pluripotent stem cells (Takahashi et al. 2006), are two exceptions to this rule.) While the external environment of the cell is certainly an important determinant of its type and function, so is its internal state which is determined by gene expression. Therefore, differentiation of a cell from a pluripotent type to a more specialized type is with no doubt guided and controlled by changes in gene expression levels.

Constructing the differentiation hierarchy for a population of cells harnessed in the lab from the human body requires first identifying the type of each cell. Traditionally, discrete cell types have been identified by scientists using a number of means, including morphology, the study of development, lineage tracing, etc. As a practical matter, however, the cell types are often identified based on their expression or non-expression of certain marker genes. Historically, these have often been various cell-surface proteins, which are fairly easy to assay biochemically. However, proteins produced from these marker genes usually have no role in the regulation of gene expression and therefore are not causative of the cell type, in the sense of being master transcriptional regulators. But then, even master transcriptional regulators need a certain cellular context to achieve their function. The true state of a cell cannot be captured fully by the expression of just a handful of genes. This raises important questions about how gene expression across the whole genome controls or reflects cell state, and in particular, differentiation hierarchies.

Billions of hematopoietic cells are produced daily from a small number of adult stem cells, which makes hematopoietic differentiation an ideal model for the study of multilineage differentiation in humans. Hematopoietic differentiation has been studied extensively not only in humans but also in mice (Spangrude et al. 1988), which has led to the isolation of different stem cell populations and the identification of multiple lineages of multipotent hematopoietic progenitors (Morrison et al. 1997). Transcriptional regulation and biological pathways in each lineage of hematopoietic differentiation have been extensively studied as well (Quesenberry et al. 2005), leading to a wide agreement on the structure of the hierarchy, which is shown in Figure 1. Most importantly, studies have shown that purified hematopoietic stem cells have a great differentiation potential that allows them to give rise to non-hematopoietic tissues (Lagasse et al. 2000 and Krause et al. 2001). Moreover, emerging evidence suggests similarities in the mechanisms that regulate self-renewal of normal stem cells and cancer cells. The evidence also suggests that tumour cells can arise from normal stem cells and may contain cancer stem cells with high proliferative potential (Reya et al. 2001). Studies of well-characterized stem cells have shown that identifying unknown stem cells using markers is not an easy task since markers that are specific for one stem cell have not been found yet (Morrison et al. 2008). Identifying hematopoietic stem cells (HSCs), however, is not as difficult since these cells are identified based on their ability to self-renew and to differentiate into cell types of all blood cell lineages upon transplantation into irradiated mice. In the past few decades, combinations of markers have been identified that allow for the identification of HSCs in bone marrow with 50% purity by flow cytometry (Kiel et al. 2005, Matsuzaki et al. 2004 and Takano et al. 2004).

Some authors have tried to reconstruct stem cell differentiation hierarchies *de novo*, based only on gene expression data. Intuitively, if we had a complete understanding of how gene expression and cell state/type were related, it should be possible to reconstruct a differentiation tree based on expression at the nodes in that tree. Kluger et al. (2004) used single linkage hierarchical clustering on microarray gene expression data of fully-differentiated human blood cells to classify them into nine distinct branches. The resulting

tree agreed with the known differentiation tree for blood cells, proving that *de novo* reconstruction of the tree is possible. Joshi et al. (2011) showed that, similar to phylogenetics, the differentiation hierarchies of mammalian tissue, specifically hematopoietic differentiation, neural differentiation and early endoderm organogenesis, can be inferred by parsimonious maximization. Their three parsimonious trees showed consistency with the current experimentally validated trees, again confirming the feasibility of *de novo* reconstruction.

We therefore became interested in testing reconstruction on the more extensive data set of Novershtern et al. (2011). This data (available at http://www.ncbi.nlm.nih.gov/geo/ with accession number GSE24759) comprises microarray gene expression data for 22215 genes in 38 distinct cell types in the hematopoiesis hierarchy (Figure 1), each assayed with between 4 and 10 replicate arrays (Table 1), for a total of 211 arrays. Importantly, the Novershtern data includes expression not just at the fully-differentiated level of the hierarchy, but at the intermediate levels as well and all the way up to the root (hematopoietic stem cells, or HSCs). The 38 cell types in the hierarchy possess varying potentials for self-renewal. The root HSCs produced in the bone marrow possess long-term self-renewal abilities and give rise to short-term self-renewing HSCs which then give rise to the progenitors that have no detectable potential for self-renewal but are more mitotically active (Morrison & Weissman 1994). The progenitors later give rise to the fully-differentiated cell types that carry out specialized functions in the blood system. Therefore, all cell types in the hematopoietic hierarchy can be isolated in a stable state from the human body and identified by looking at the expression of a combination of marker genes (Spangrude et al. 1988 and Weissman 1994). The majority of the cell types in the differentiation tree shown in Figure 1 were purified from umbilical cord blood, a source enriched in undifferentiated populations. However, the T cells (T1-8), the B cells (B1-4), the NK cells (N1-4) and dendritic cells (PLD and MLD) were purified from adult peripheral blood because these terminally differentiated populations require exposure to antigens after birth. Samples from four to seven distinct donors were purified for each population using multiparameter flow cytometry (See Novershtern et al. 2011 for more details). For the stem cells (HSC1 and HSC2), the

progenitors (CMP, MEP and GMP) and the erythrocytes (E1-5), cell types were identified using antibodies against the marker genes listed beside each cell type in Figure 1 followed by flow cytometry for labelled antibodies. The remaining cell types were identified using flow scatter properties as well as antibodies against the marker genes listed beside each cell type in Figure 1. A list of all marker genes used for cell type identification is also shown in Table 1.

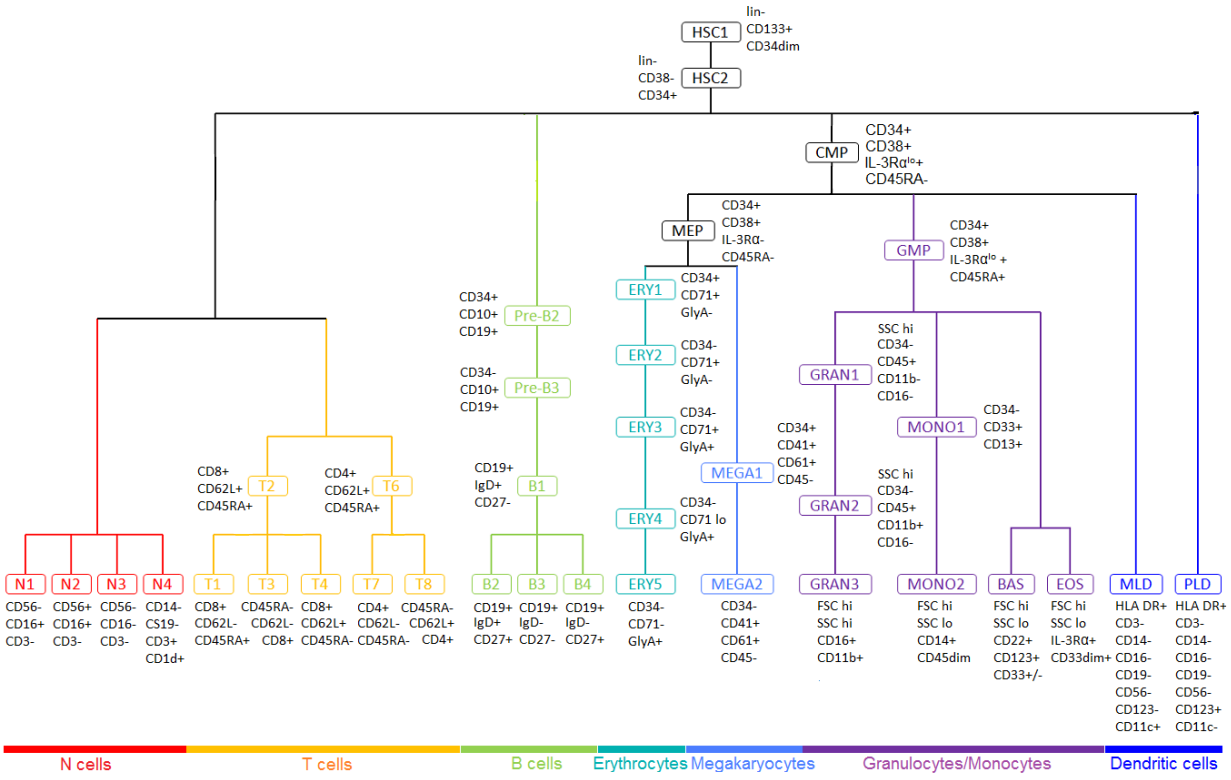


Figure 1 Human Hematopoietic Differentiation Tree
The 38 cell types in the hematopoietic differentiation hierarchy: Hematopoietic stem cells (HSC1-2), common myeloid progenitor (CMP), megakaryocyte/erythroid progenitor (MEP), erythroid cells (ERY1-5), CFU-MK (MEGA1), megakaryocytes (MEGA2), Granulocyte/monocyte progenitor (GMP), CFU-G (GRAN1), neutrophilic metamyelocyte (GRAN2), neutrophil (GRAN3),CFU-M (MONO1), monocytes (MONO2), eosinophil (EOS), basophil (BAS), myeloid dendritic cell (MLD), plasmacytoid dendritic cell (PLD), early B cell (Pre-B2), pre-B cell (Pre-B3), naive B cell (B1), mature B cell class able to switch (B2), mature B cell (B3), mature B cell class switched (B4), mature NK cell (N1-4), naive CD8+ T cell (T2), CD8+ effector memory RA (T1), CD8+ effector memory (T3), CD8+ central memory (T4), naive CD4+ T cell (T6), CD4+ effector memory (T7), and CD4+ central memory (T8). Near each cell type is a list of the marker genes whose expression was used by the authors of Novershtern et al. (2011) to identify that cell type. A plus/minus sign following a marker gene name indicates a relatively higher/lower expression level of that gene in a cell type.

Cell Populations	Abbreviations	Samples	Marker Genes
Hematopoietic Stem Cells			
Hematopoietic Stem Cell 1	HSC1 or HS1	10	lin-, CD133+, CD34dim
Hematopoietic Stem Cell 2	HSC2 or HS2	4	lin-, CD38-, CD34+
Myeloid Progenitors			
Common Myeloid Progenitor	CMP	4	CD34+, CD38+, IL-3Rα^{lo}+, CD45RA-
Megakaryocyte/Erythroid Progenitor	MEP	9	CD34+, CD38+, IL-3Rα-, CD45RA-
Granulocyte/Monocyte Progenitor	GMP	4	CD34+, CD38+, IL-3Rα^{lo}+, CD45RA+
Erythrocytes			
Erythroid 1	ERY1 or E1	7	CD34+, CD71+, GlyA-
Erythroid 2	ERY2 or E2	7	CD34-, CD71+, GlyA-
Erythroid 3	ERY3 or E3	6	CD34-, CD71+, GlyA+
Erythroid 4	ERY4 or E4	7	CD34-, CD71 lo, GlyA+
Erythroid 5	ERY5 or E5	6	CD34-, CD71-, GlyA+
Megakaryocytes			
Colony Forming Unit Megakaryocyte (CFU -MK)	MEGA1 or MG1	5	CD34+, CD41+, CD61+, CD45-
Megakaryocyte	MEGA2 or MG2	7	CD34-, CD41+, CD61+, CD45-
Granulocytes			
Colony Forming Unit Granulocyte (CFU-G)	GRAN1 or G1	5	CD34-, CD45+, CD11b-, CD16-
Neutrophilic Metamyelocyte	GRAN2 or G2	4	CD34-, CD45+, CD11b+, CD16-
Neutrophil	GRAN3 or G3	4	CD11b+, CD16+
Monocytes			
Colony Forming Unit Monocyte	MONO1 or MN1	4	CD34-, CD33+, CD13+
Monocyte	MONO2 or MN2	5	CD14+, CD45dim
Basophil	BAS	6	CD22+, CD123+, CD33 +/-
Eosinophil	EOS	5	IL-3Rα+, CD33dim+
B Lymphoid Progenitors			
Early B Cell	Pre-B2 or PB2	4	CD34+, CD10+, CD19+
Pro B Cell	Pre-B3 or PB3	5	CD34-, CD10+, CD19+
B Cells			
Naive B Cell	B1	5	CD19+, IgD+, CD27-
Mature B Cell class able to switch	B2	5	CD19+, IgD+, CD27+
Mature B Cell	B3	5	CD19+, IgD-, CD27-

Mature B Cell class switched	B4	5	CD19+, IgD-, CD27+
Dendritic Cells			
Plasmacytoid Dendritic Cell	PLD	5	HLA DR+, CD3-, CD14-, CD16-, CD19-, CD56-, CD123+, CD11c-
Myeloid Dendritic Cell	MLD	5	HLA DR+, CD3-, CD14-, CD16-, CD19-, CD56-, CD123-, CD11c+
T Cells			
Effective Memory RA CD8+ T Cell	T1	4	CD8+, CD62L-, CD45RA+
Naive CD8+ T Cell	T2	7	CD8+, CD62L+, CD45RA+
Effective Memory CD8+ T Cell	T3	6	CD8+, CD62L-, CD45RA-
Central Memory CD8+ T Cell	T4	7	CD8+, CD62L+, CD45RA-
Naive CD4+ T Cell	T6	7	CD4+, CD62L+, CD45RA+
Effective Memory CD4+ T Cell	T7	7	CD4+, CD62L-, CD45RA-
Central Memory CD4+ T Cell	T8	7	CD4+, CD62L+, CD45RA-
Natural Killer Cells			
Mature NK Cell 1	N1	4	CD56-, CD16+, CD3-
Mature NK Cell 2	N2	5	CD56+, CD16+, CD3-
Mature NK Cell 3	N3	5	CD56-, CD16-, CD3-
NKT	N4	4	CD14-, CD19-, CD3+, CD1d+

Table 1 Number of Gene Expression Samples and List of Marker Genes for each Cell Type
Gene expression replicates for all 38 cell types were taken from Novershtern et al (2011) and are available at http://www.ncbi.nlm.nih.gov/geo/ with accession number GSE24759. In the fourth column is a list of the marker genes whose expression was used by the authors of Novershtern et al. (2011) to identify each cell type. A plus/minus sign following a marker gene name indicates a relatively higher/lower expression level of that gene in a cell type.

The hierarchical clustering approach espoused by Kluger et al. (2004) and the maximum parsimony approach of Joshi et al. (2011) produce trees where all cell types, including those that are not fully-differentiated, are classified as leaves. Although we were not satisfied with the trees constructed by these two approaches, they still bear some relationship to the correct differentiation tree. We present the results of these two approaches in Chapter 3 and we show how they relate to the correct tree but do not accurately reconstruct it, thus leaving us dissatisfied and seeking another approach. Other traditional approaches to understanding

gene expression patterns across multiple conditions, such as principal components analysis or K-means clustering, can group cell types based on gene expression, but they do so without knowledge of the differentiation hierarchy. Reconstructing a differentiation tree from gene expression data alone without any prior knowledge of the tree structure is a very hard task. However, reconstructing differentiation trees in a supervised setting can help in reaching the ultimate goal of reconstructing trees in an unsupervised setting. Moreover, a method that is able to acquire knowledge from a tree with a predetermined structure offers two main advantages over the traditional unsupervised methods. First, it provides more knowledge of what genes control or relate to the structure of the differentiation hierarchy, and from those genes many other conclusions may be reached regarding the biological processes and functions that are active throughout differentiation. Second, a supervised method can offer more predictive power in a sense that it can be used to induce differentiation trees from other gene expression data. Therefore, one of our goals in this thesis is to explain and analyze the relationship between gene expression in different cell types and the structure of the differentiation hierarchy. The second goal is to learn a distance metric that allows for tree reconstruction in a supervised setting and also performs gene selection that helps us identify genes that are most relevant to the structure of the tree. The third goal is to provide another contribution towards solving the more complicated problem of predicting differentiation trees either based on gene expression data alone with no prior knowledge of the tree structure or based on knowledge acquired from other hierarchies of differentiation.

This thesis makes three main contributions towards understanding the relationships between gene expression and stem cell differentiation hierarchies. In chapter 3, we perform an analysis on the Novershtern data to better understand gene expression patterns in different parts of the hematopoiesis differentiation tree. We propose one parametric model and another statistical model that give each gene one of the four labels: up-regulated, down-regulated, fluctuating and stable in each lineage of the tree. Our results show that besides marker genes there are thousands of other genes whose expression data can be used to discriminate between different branches of the differentiation tree of Figure 1. We then apply the hierarchical clustering approach espoused by Kluger et al. (2004) and the maximum

parsimony approach of Joshi et al. (2011) to the Novershtern data and we illustrate how they fail to produce the proper type of tree we seek, although they do give some useful insights into how gene expression in different cell types relates to the structure of their differentiation hierarchy. We also show that although a minimum spanning tree of all the cell types is a proper type of tree, the minimum spanning trees constructed using several distance metrics on the Novershtern data do not perfectly match with the correct differentiation tree.

In chapter 4, we propose two novel approaches to solving the following problem: Given the differentiation hierarchy and gene expression data at each node, construct a weighted Euclidean distance metric such that the minimum spanning tree with respect to that metric is precisely the given differentiation hierarchy. The first approach works by first deciding what distance between every two nodes in the tree we aim to reach such that the differentiation tree with its given structure is a minimum spanning tree of its nodes. We then translate these pairwise distances to a set of linear equations and propose a mean square error minimization (MSE) framework that aims to find a weight vector that satisfies these equations. In the second approach we provide a set of linear inequality constraints that are provably sufficient for the desired construction, but this time allowing for more flexibility in the pairwise distances between nodes. We then propose a linear programming framework to identify sparse sets of weights, effectively identifying genes that are most relevant for discriminating different parts of the tree.

In chapter 5, we try the MSE approach on the Novershtern data for the differentiation tree in Figure 1 using an arbitrary choice of the targeted pairwise distances but do not succeed in finding a weight vector that allows for tree reconstruction. Although other choices of pairwise distances might work, we choose to save the effort and apply the second more promising linear programming approach and we succeed in constructing a weighted Euclidean metric that uses just 175 genes. These 175 genes are different than the marker genes highlighted in Figure 1 and Table 1 hence offering a novel alternative way of discriminating different branches of the tree. However, we find that there are many alternative sets of weights that satisfy the linear constraints. Thus, in the style of random-

forest training, we also construct metrics based on random subsets of the genes and compare them to the metric of 175 genes and then report on the selected genes and their biological functions. Demonstrating the predictive power of our method by learning a distance metric from multiple data sets and then measuring its power in predicting other trees from other data sets is of high interest to us. However, finding other reliable and rich gene expression data sets and carrying out the validation on our method would require a fairly large amount of time and effort which exceeds the requirements of a master's thesis given the amount of work we have already done. Therefore, we focus more on the biological explanatory benefits of our approach which are impressive enough. However, we choose to further evaluate the linear program on synthetic data with a test that is less time consuming by testing how often it can identify which columns were used to construct minimum spanning trees from random matrices of variable sizes. Although the program is not designed specifically for this purpose, it still succeeds in identifying the correct columns for matrices of specific sizes. We conclude in chapter 6 by discussing potential applications of our method and possible ways to expand on our work in the future.

2 BACKGROUND

Although, up to our knowledge, no one has yet tried to learn a distance metric that allows for the reconstruction of a differentiation tree from gene expression data by selecting a few number of genes, a large amount of work has been done in multiple areas that are related to our work such as studies of regulatory circuits in stem cell differentiation, dimensionality reduction on gene expression data and distance metric learning. Therefore, we start this chapter by discussing relevant work done by others in section 2.1 and we also discuss how they do not completely address the same problem we address. In this thesis, we also make use of several computational tools that require some background knowledge in more than one area. For example, in Chapter 3, we use a t-test to evaluate the statistical significance of differential gene expression between different cell types. We therefore briefly explain in section 2.2 the purpose of using statistical significance testing. We then provide a general introduction to distance metrics in section 2.3 and give a definition for each metric we later use in the other chapters. Since we also demonstrate in Chapter 3 how hierarchical clustering and maximization of parsimony do not produce proper trees that allow for reconstruction of the differentiation tree, we explain in sections 2.4 and 2.5 how these two methods work using simple examples. We also explain how a heatmap can help us extract information from data matrices since we use this tool to present many of our results in Chapter 3 and Chapter 5. Our proposed method in Chapter 4 revolves around reconstructing the differentiation tree as a minimum spanning tree within a linear programming framework. Therefore, in section 2.6, we describe and illustrate with a simple example what a minimum spanning tree is and we end the chapter with a brief introduction to linear programming in section 2.7.

2.1 Related Work

Regulation of stem cell differentiation has been investigated in many studies (See for example Aplan et al. 1992, Segal et al. 2003, Akashi 2005, Boyer et al. 2005, Bakker et al. 2007, Ng et al. 2007 and Rosenbauer et al. 2007). Novershtern et al. (2011) identified modules of co-expressed genes most of which are expressed across multiple lineages of hematopoietic differentiation. They also found interconnected cis-regulatory and transcriptional circuits that control cell state, suggesting a more complex hematopoietic

regulatory system whose major aspects are still unknown. Iwasaki & Akashi (2007) showed that hematopoietic differentiation is controlled by a small number of transcription factors that are expressed in a specific lineage and interact with each other to control cell fate decisions. Other studies, however, have shown that a larger number of transcription factors are involved in more complex circuits of regulation in immune cell types (Amit et al. 2009 and Suzuki et al. 2009) and other stem cell populations (Müller et al. 2008 and Davidson 2001). However, these studies focus on understanding transcription factor regulation in stem cell differentiation but do not explicitly try to reconstruct the differentiation tree or model the pairwise relationships between different cell types in a tree.

Several standard clustering and machine learning tools are available for the computational analysis of microarray gene expression data (Quackenbush 2001). Many studies have made use of these methods to relate gene expression to stem cell differentiation. For example, Brunet et al. (2004) used nonnegative matrix factorization to identify molecular and gene expression patterns by reducing the dimension of expression data from thousands of genes to a small number of metagenes. Müller et al. (2008) used the method proposed by Brunet et al. to categorize a collection of ~150 samples of pluripotent, multipotent and differentiated cell types. They were able to separate pluripotent cell types from other cell types and uncover a protein-protein network that is shared by pluripotent cells using further bioinformatic analysis. However, none of these papers try to explicitly reconstruct the differentiation hierarchy or show that it is possible with their approaches. Luo et al. (2004) proposed a dynamically growing self-organizing tree algorithm that overcomes some of the drawbacks of traditional hierarchical clustering methods such as fixed topology structure and misclustered data that cannot be re-evaluated. However, the tree structure produced by this dynamic method is still a dendrogram that places all nodes as leaves. Principal components analysis (PCA) has also been used to cluster cells at different differentiation levels from different body tissues based on gene expression. Examples of such work are Sharov et al. (2003 and 2005), Matoba et al. (2006) and Aiba et al. (2006 and 2009). However, PCA aims to separate cell types by transforming their coordinates (genes) to a set of orthogonal variables that account for as much of the variability in the data as possible, but it does so

without knowledge of the structure of the differentiation tree. Therefore, PCA can give insights into the overall structure of the tree by clustering cell types together but it does not reconstruct the tree nor does it relate cell types together based on their relative positions in the tree. The only two papers we found that explicitly try to reconstruct differentiation trees from gene expression data are Kluger et al. (2004) and Joshi et al. (2011). Kluger et el. (2004) used single linkage hierarchical clustering to separate the human blood cell types into nine distinct lineages. However, their data consisted of gene expression of fully-differentiated cell types only and their resulting tree does not give any description of the internal structure for each lineage. It is therefore not clear from their work whether hierarchical clustering can reconstruct a differentiation tree with intermediate cell types. Similarly, the trees produced by Joshi et al. (2011) using parsimony maximization consisted of fully-differentiated cell types only. One drawback of both hierarchical clustering and maximum parsimony is that all clustered cell types, intermediate and fully-differentiated, are placed as leaves in the resulting trees. Another drawback of both methods is that they also do not make use of any prior knowledge of the correct tree structure.

On the other hand, our proposed method makes use of our prior knowledge of the differentiation hierarchy to learn a weighted Euclidean distance metric that allows for tree reconstruction and selects a very small number of genes to participate in the distance measure. Many approaches for learning a distance metric have been proposed in the machine learning literature. For instance, Xing et al. (2003) introduced a convex optimization method for learning a distance metric that minimizes the distance between objects of the same class subject to linear constraints that ensure objects from different classes are well separated. Similar work was done by Kwok et al. (2003) and Globerson et al. (2005). Weinberger et al. (2006, 2008) and Ying et al. (2012) aim to improve K-nearest neighbour classification by learning a Mahalanobis distance metric from labelled training examples. Mahalanobis distance is a multi-dimensional measure of how far a point is from the mean of some distribution. It can also be a distance measure between two random vectors $\vec{x}$ and $\vec{y}$ of the same distribution with a covariance matrix C. If the covariance matrix is diagonal, then the resulting distance measure is a weighted Euclidean distance where each term $(x_i - y_i)^2$ is

normalized by the variance of the distribution from which x_i and y_i were drawn. A comprehensive survey of distance metric learning algorithms can also be found in Yang & Jin (2006). However, all of these methods focus on improving performance in classification tasks, usually binary classification. Nevertheless, we were inspired by these works to develop our own method in Chapter 4 for learning a distance metric whose main purpose is to reconstruct a differentiation hierarchy.

2.2 Statistical Significance Testing

Experimental measurements are always subject to human and machine error, which gives rise to discrepancies that are difficult to quantify and remove from measured data. Therefore, experiments are usually repeated multiple times and replicate measurements are obtained as a way to quantify that error by taking into consideration variances in multiple readings. Biological data is no exception and in fact can be very misleading if its noise component is not properly handled. Each of the 38 cell types in the tree of Figure 1 is represented by multiple replicates in the Novershtern data (See Table 1). These replicates surely vary for each cell type and their variances need to be considered in some cases, such as in our search for genes that discriminate different parts of the hematopoiesis differentiation tree in Chapter 3. When measuring differences in gene expression between cell types we need to confirm whether these differences are statistically significant and not merely a coincidence. Many tests are available for addressing the statistical significance of differences measured between two data samples. Although the validity of these tests has been a topic of debate in the statistics community (Schmidt 1996 and Harlow et al. 2013), they nevertheless can be used to at least strengthen our confidence in the significance of our measurements for differential gene expression. Therefore, whenever we need to assess the statistical significance of differential gene expression we use the unpaired t-test, a widely used parametric type of Null-Hypothesis statistical testing.

2.3 Distance Metrics

One of the major contributions in this thesis is finding a distance metric that allows for reconstruction of the differentiation tree. A distance metric is a mapping $d: X \text{ x } X \rightarrow R_0^+$ over a vector space X that satisfies the following axioms for all vectors $x_i,\ x_j,\ x_k\ \in X$

$d(x_i, x_j) \geq 0$ (Non-negativity)

$d(x_i, x_j) = d(x_j, x_i)$ (Symmetry)

$d(x_i, x_j) + d(x_j, x_k) \geq d(x_i, x_k)$ (Triangle Inequality)

$d(x_i, x_j) = 0 \leftrightarrow x_i = x_j$ (Distinguishability)

However, there are mappings that do not satisfy all four axioms of a metric but are still useful in many situations. This leads to other definitions of generalized metrics that are defined by relaxing one or more axioms of a metric. For example, a pseudometric is a mapping that satisfies the first three axioms of a metric but does not satisfy the axiom of distinguishability. In other words, $d(x_i, x_j) = 0$ is allowed in pseudometrics for distinct vectors $x_i \neq x_j$, however $d(x_i, x_i) = 0$ for each single vector x_i remains a requirement. A quasimetric is a mapping that satisfies the axioms of a metric with the possible exception of the axiom of symmetry. A semimetric is a mapping that satisfies the axioms of a metric except for the triangle inequality axiom. Other generalized metric definitions also exist including those with mixed prefixes such as a pseudoquaismetric. However, we do not go into the details of generalized metrics since they are beyond the scope of this thesis.

For the purpose of showing how hierarchical clustering fails to produce the proper type of tree we only use a few standard distance metrics, namely Euclidean, L1, Cosine, Chebychev and Correlation distances. Other metrics that are meant for specific classification problems would need to be adapted to the new data and problem we are dealing with before using them, a task that is unnecessary for our purpose. We also use the hamming distance metric in Chapter 3 to cluster cell types based on gene labels we assign using our two proposed models. Therefore, we give a definition of each of these standard metrics on two vectors

$x_i = (x_{i1}, x_{i2}, \ldots\ldots, x_{im})$ and $x_j = (x_{j1}, x_{j2}, \ldots\ldots, x_{jm})$ below.

Euclidean distance: $$d(x_i, x_j) = (x_i - x_j)(x_i - x_j)^T = \sqrt{\sum_{k=1}^{m} (x_{ik} - x_{jk})^2}$$

L1 distance: $$d(x_i, x_j) = \sum_{k=1}^{m} |x_{ik} - x_{jk}|$$

Cosine distance: $$d(x_i, x_j) = 1 - \frac{x_i x_j^T}{\sqrt{(x_i x_i^T)(x_j x_j^T)}}$$

Chebychev distance: $$d(x_i, x_j) = \max_k \{|x_{ik} - x_{jk}|\}$$

Hamming distance: $$d(x_i, x_j) = \frac{\#(x_{ik} \neq x_{jk})}{m}$$

Correlation distance: $$d(x_i, x_j) = 1 - \frac{(x_i - \bar{x_i})(x_j - \bar{x_j})^T}{\sqrt{(x_i - \bar{x_i})(x_i - \bar{x_i})^T}\sqrt{(x_j - \bar{x_j})(x_j - \bar{x_j})^T}}$$

where $\bar{x_i} = \frac{1}{m}\sum_{k=1}^{m} x_{ik}$ and $\bar{x_j} = \frac{1}{m}\sum_{k=1}^{m} x_{jk}$

2.4 Hierarchical Clustering

Hierarchical clustering (Murtagh 1983) is a technique that has traditionally been used in tree reconstruction, and we also use in our analysis of the Novershtern gene expression data. Hierarchical clustering, also known as nearest neighbor joining, organizes a group of nodes into a hierarchy of clusters. Given a set of nodes N and a matrix $M \in \mathbf{R}^{n \times m}$ where each row in M corresponds to a feature vector of a node in N, and given a node-to-node distance metric d, hierarchical clustering starts by considering each node to be a cluster on its own and calculates the distance between each pair of clusters in N using the distance metric d. The two clusters with the smallest pairwise distance are merged into one larger cluster which replaces those two clusters in N. The pairwise distances are then re-calculated for the new

set N and merging is repeated until all nodes/clusters are hierarchically merged into a single large cluster. Three common linkage methods for calculating the distance $d(A, B)$ between two clusters A and B are:

Complete linkage: $d(A, B) = \max\{d(a, b): a \in A, b \in B\}$

Single linkage: $d(A, B) = \min\{d(a, b): a \in A, b \in B\}$

Average linkage: $d(A, B) = \frac{1}{|A||B|}\sum_{a\in A}\sum_{b\in B} d(a, b)$

The results of hierarchical clustering are usually presented in a tree-like structure called a dendrogram. Heatmaps make use of hierarchical clustering to better convey information stored in a matrix using colour patterns. Given a matrix $M \in \mathbf{R}^{n\times m}$, a heatmap presents the entries of M using a colour scale which allows for better visualization. Hierarchical clustering with any linkage method and distance metric can be applied on the rows or columns (or both) of M and the heatmap then arranges its rows and columns in the same order they appear in their dendrograms. Any similar patterns between different rows or different columns of M can then be easily identified in the heatmap by looking at areas of large concentrations of a specific colour. For example, assume a set of nodes $N = \{a, b, c\}$ with their respective feature vectors (2, -2, 3, 1), (-2, -2, 3, -3) and (1, 2, 1, 3) stored in a matrix M. If we apply hierarchical clustering with average linkage to both rows and columns of M using Euclidean distance we obtain the heatmap shown in Figure 2, where it is clear that nodes a and b are more similar to each other than node c is, and that features 1 and 2 show similar patterns in all nodes. Although these patterns can be seen directly from the feature vectors of the nodes, when dealing with thousands of features such as in the case of gene expression data, heatmaps are necessary to view such patterns. In later chapters, we use average linkage every time we apply hierarchical clustering to the rows and/or columns of a heatmap. However, when applying hierarchical clustering to the Novershtern data in Chapter 3, we use all three linkage methods (single, average and complete linkage) to study their impact on the structure of the produced tree.

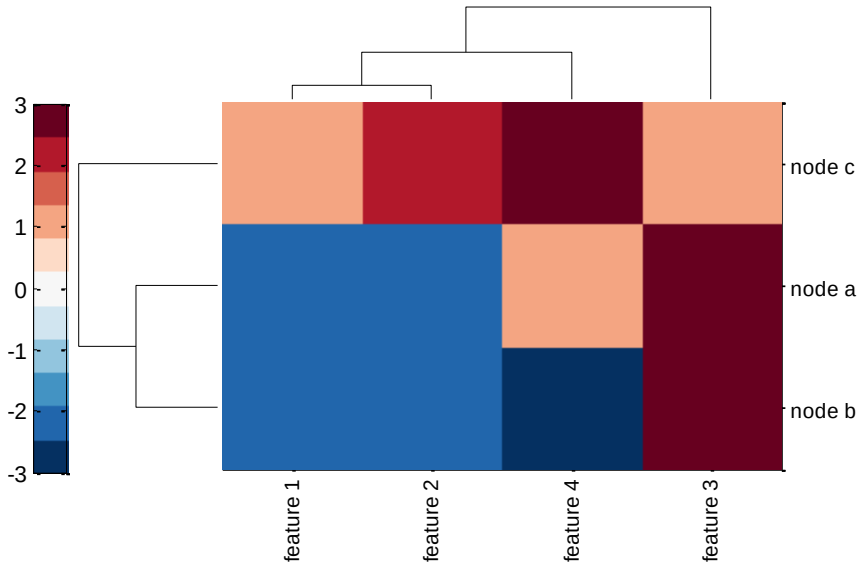


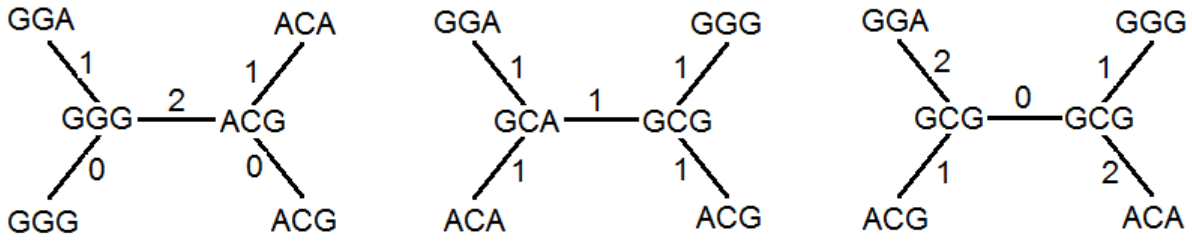
Figure 2 Example Heatmap
Heatmap generated by applying hierarchical clustering with average linkage using Euclidean distance on three nodes a, b and c with the three feature vectors (2, -2, 3, 1), (-2, -2, 3, -3) and (1, 2, 1, 3) respectively.

2.5 Maximization of Parsimony

Maximization of parsimony, which is to prefer the simpler of two otherwise equally adequate hypotheses, has been proven useful in many scientific and mathematical fields and was first presented by Walter M. Fitch (1971). The idea of parsimonious maximization stems from Occam's razor, a principle of theoretical parsimony suggested by William of Ockham in the 1320s, which asserted that it is vain to give an explanation that includes more assumptions than necessary. One of its main applications is in computational phylogenetics, where the goal is to construct a tree that shows the evolutionary relationships and ancestries for a group of species. In phylogenetics, each species is associated with a feature vector that consists of the consensus sequence of nucleotide bases in its genome, each base being one of the following four: adenine (A), cytosine (C), guanine (G), or thymine/uracil (T/U). Despite their limitations, phylogenetic trees provide many insights into the evolution of species, and Joshi et al. (2011) has shown that one of the methods used to infer these trees, maximization of parsimony, can also be used to infer stem cell differentiation trees.

Maximum parsimony is a character-based method that searches for a tree that assumes the least change or distance between its nodes. Because the method is character-based, distance between two nodes is measured by counting the number of positions in which the two compared feature vectors differ. When the feature vectors are nucleotide sequences, each position in the sequence can take one of the four characters A, C, G and T. In the simplest case, it may be assumed that a change from one character to another happens in one step thus the distance between any two different characters is 1. In this case, parsimonious maximization would be similar to using hamming distance with hierarchical clustering where the distance between two objects is the ratio of the number of positions in which their feature vectors differ to the total number of features in the vector. However, in other cases, other assumptions may need to be made when building a parsimonious tree. For example, it can be assumed that a change from one character to another cannot happen in one step but rather in two steps, such as an A changing to C requiring that first A must change to T and then from T to C. In that case, transitions between characters are given weights that reflect the probability of each transition occurring. In this case, the distance measure is not the hamming distance anymore.

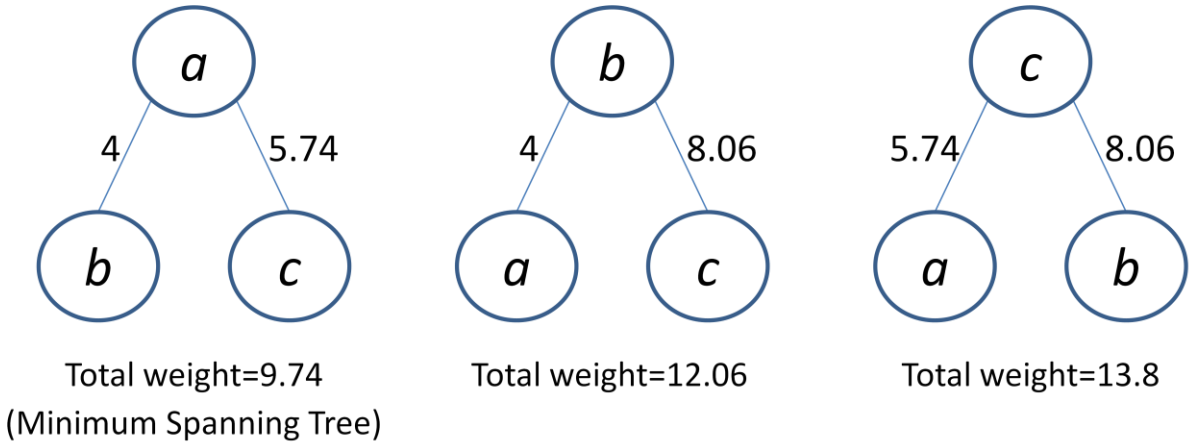
A common property between constructing a tree with maximum parsimony and hierarchical clustering is that both methods create intermediate nodes in the constructed tree. For each two nodes/sequences, maximum parsimony assumes a common ancestor node/sequence that minimizes the distance or change between that ancestor and each of the two nodes. The maximally parsimonious tree is the tree that minimizes the total sum of distances on all edges. For example, given four sequences GGA, GGG, ACA and ACG, the only three possible non-rooted trees we could build from these sequences are shown below with their assumed intermediate sequences. The maximally parsimonious tree for these four sequences would be the first tree on the left side.



However, if we want to use maximum parsimony to build a tree for a group of objects with real-valued feature vectors, such as gene expression, we first need to digitize the data into discrete states. The number of states we choose may potentially have an effect on which tree happens to have the least number of changes. Therefore, applying maximum parsimony on real-valued data requires making assumptions on the number of states used and the cut-off values that determine these states. Since Joshi et al. used only two states (0 and 1) to reconstruct their differentiation trees but with different cut-off values, we use the same number of states and same cut-off values to test their approach on our data set.

2.6 Minimum Spanning Tree

As we describe later in Chapter 4, our approach to reconstructing the differentiation tree is based on the concept of a minimum spanning tree (Graham & Hell 1985). Given a connected undirected graph G with nodes N and weighted edges E, a minimum spanning tree (MST) is a tree that spans all nodes of N while minimizing the total sum of weights on all its edges. For example, assume the same set of nodes $N = \{a, b, c\}$ we used in the hierarchical clustering example with their respective feature vectors (2, -2, 3, 1), (-2, -2, 3, -3) and (1, 2, 1, 3). If we assume the weight for each pair of nodes in N to be their Euclidean distance then the pairwise weights would be $d(a, b) = 4$, $d(a, c) = 5.74$ and $d(b, c) = 8.06$. The three trees shown below are all possible trees we can construct for these three nodes. The first tree with a minimal total weight of 9.74 is the minimum spanning tree of the nodes of N.



Prim's algorithm (Prim 1957) is one of the algorithms that find a minimum spanning tree for a connected weighted undirected graph. Despite being greedy, this algorithm is guaranteed to produce a minimum spanning tree through the following steps:

- Input: Non-empty connected graph with nodes N and weighted edges E
- Initialize $N_{MST} = \{n\}$ where n is an arbitrary node from N, and initialize $E_{MST} = \{\}$
- Repeat until $N_{MST} = N$
  - Choose a node u from N_{MST} and a node v from N that is not in N_{MST} such that edge $\{u, v\}$ has minimal weight
  - Add v to N_{MST} and $\{u, v\}$ to E_{MST}
- Output: Minimum spanning tree with nodes N_{MST} and edges E_{MST}

2.7 Linear Programming

Our approach to finding a distance metric that allows for reconstruction of the differentiation tree as a minimum spanning tree boils down to solving a linear program (LP), a convex optimization problem that optimizes a linear function subject to linear inequality and equality constraints. A linear program can generally be presented in the following form

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, n \\ & h_i(x) = 0, \quad i = 1, \ldots, p \end{array}$$

to describe the problem of finding a solution to the optimization variable x that minimizes the affine objective function $f_0(x)$ among all $x \in \mathbf{R}^m$ that satisfy the affine inequality constraints $f_i(x) \leq 0, \quad i = 1, \ldots, n,$ and the affine equality constraints $h_i(x) = 0, \quad i = 1, \ldots, p.$

The domain D of the optimization problem is the set of points for which the objective function f_0 and all constraint functions f_i and h_i are defined.

$$D = \bigcap_{i=0}^{n} \mathbf{dom}\, f_i \cap \bigcap_{i=1}^{p} \mathbf{dom}\, h_i$$

The optimization problem is said to be feasible if there exists at least one feasible point $x \in D$ that satisfies the constraints $f_i(x) \leq 0$ for $i = 1, \ldots, n,$ and $h_i(x) = 0$ for $i = 1, \ldots, p,$ otherwise it is said to be infeasible.

Since the objective function and constraints of the LP are linear, the program can be written in the following form

$$\begin{array}{ll} \text{minimize} & c^T x + d \\ \text{subject to} & Gx \leq h \\ & Ax = b \end{array}$$

where $G \in \mathbf{R}^{n \times m}$ and $A \in \mathbf{R}^{p \times m}$.

The problem would still be a linear program if we were to maximize $c^T x + d$ since that would be similar to minimizing $-c^T x - d$ which is still a linear objective. It is also common to omit the constant d in the objective function since any feasible point that minimizes $c^T x + d$ will also minimize $c^T x$. Several algorithms have been developed to solve linear programs in polynomial time including the two basic exchange algorithms: simplex and Criss-cross. Other interior point methods were developed such as the ellipsoid algorithm, projective algorithm and path-following algorithm. However, efficiency of these methods and the structure of solutions they generate can vary from one LP problem to another.

An integer linear program (ILP) is a linear program in which some or all the variables in the feasible set are restricted to be integers. A naive way of solving an ILP is through LP relaxation, which is done by removing the restriction that the variables in x must be integers and then solving the corresponding LP. The entries of the solution to the relaxed LP are then rounded to integers. However, the solution may not remain optimal, or even feasible, after it is rounded. Therefore, other methods are usually needed to find an exact solution to an ILP such as the branch-and-bound method and its variants. Heuristic methods such as hill climbing and simulated annealing are also available to solve ILPs that are intractable due to the NP-completeness of ILP problems (Papadimitriou 1981 and Garey et al. 2002). However, unlike LPs that are usually solved in polynomial time, the complexity of solving ILPs is usually exponential and optimality is not always guaranteed for the solutions they generate. More on the advancements made in linear and integer programming can be found in Beasley et al. 1996. We solve the LP we propose in Chapter 4 using the lp_solve program (Berkelaar et al. 2007) that implements the revised simplex method for LPs and the branch-and-bound method for ILPs.

3 A DESCRIPTIVE ANALYSIS OF THE HEMATOPOIESIS DIFFERENTIATION TREE

The number of marker genes used to identify different cell types in the tree of Figure 1 is extremely small compared to the number of genes (22215) available in the data set we have in our hands. Although observing the expression of these marker genes in all parts of the tree is important, it is also important that we explore the expression of all other genes to get a clear understanding of the nature of the data we are dealing with. In this chapter, we first take a look at the expression of the marker genes highlighted in Figure 1 and we see how their expression relates to the structure of the differentiation tree. We then propose two models that allow us to label genes in each lineage as either up-regulated, down-regulated, fluctuating or stable. We then show that distinct lineages in the tree can be discriminated using the labels assigned to the genes by these two models. Although we use them for labelling genes so that we can understand gene behaviour in different branches of the tree, the two models we propose can also be used for gene selection. We also show that traditional methods such as hierarchical clustering and maximization of parsimony cannot reproduce the differentiation tree of Figure 1 because they classify all cell types as leaves. On the other hand, a minimum spanning tree offers a plausible way to reconstruct the differentiation hierarchy if the correct distance metric is used. Using our two models, we find thousands of genes that are up-regulated, down-regulated or fluctuating in at least one lineage. Hence, all these genes are potentially important for tree reconstruction. Therefore, we decide to give each gene in the data set the chance to participate in the distance metric learning task. However, our method will need to select the fewest number of genes that will participate in that distance metric. In other words, our proposed tree construction method does not only need to learn a distance metric but also needs to perform gene selection as well.

The data set we obtained from the Novershtern paper consists of gene expression data for 38 hematopoietic cell types. Each cell type is associated with multiple replicates each of them containing expression values for 22215 gene probes. Therefore we combine the replicates of all cell types into a matrix of 211 rows and 22215 columns. A second matrix of data was created by averaging the expression of each gene across all replicates of a cell so that each

cell is represented by one array. Now that each cell is associated with only one gene expression profile, the new data matrix has 38 rows only and 22215 columns. Some computational methods can be sensitive to data normalization (Shanker et al. 1996, Sola et al. 1997, Kim et al. 2006 and Hoffmann et al. 2002). PCA, for example, tends to assign larger weights to variables of larger variances even if the difference in variances is due to using different measuring scales for different variables. We therefore normalize our data in the second matrix so that the mean expression of each gene over all cells is 0 with a standard deviation of 1. Our choice of which data matrix to use depends on what we aim to achieve. For example, to measure the statistical significance of the differential expression of a gene between two cell types we need to take into account its expression in all replicates of both cell types. On the other hand, to construct trees using existing approaches as well as the approaches we propose in chapter 4 it is more convenient to represent each cell type by the mean of all its replicates. We also use the mean of all replicate arrays whenever we display expression of any group of genes. From this point and on, we refer to the cell types by the names listed in Table 1, which may be shorter than the names appearing in the tree of Figure 1. The shorter names are more convenient for displaying our results, especially in tree form.

We first looked into expression of the marker genes highlighted in Figure 1. Since IL-3RA and CD123 are just two different names for the same marker gene we treat them as one gene. The same applies to CD45 and its isophorm CD45RA. The two terms FSC (Forward Scatter Cytometry) and SSC (Side Scatter Cytometry) listed beside the four cell types G3, MN2, EOS and BAS indicate the results of flow cytometry, a technique used to measure physical and chemical characteristics of cells such as diameter, volume, surface area and most importantly surface proteins. Therefore, although these terms indicate the presence or absence of certain biological markers located inside a cell and on its membrane, they are not marker genes themselves and thus no expression data is associated with them. As a result we were left with 28 marker genes listed on the tree and were able to identify 70 probe sets in the Novershtern data that corresponded to 27 of these 28 genes. Expression of these 27 genes (each averaged over all its probe sets) is shown in Figure 3. The only gene we were not able to locate in the Novershtern data, thus not included in the heatmap of Figure 3, is IgD.

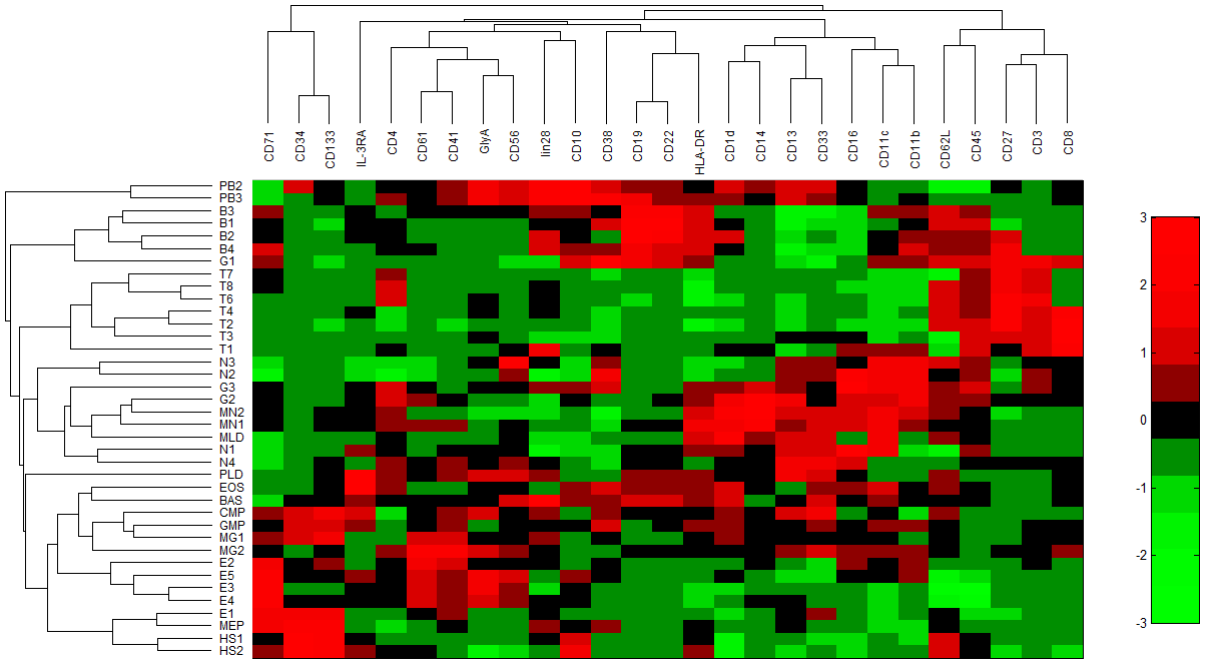


Figure 3 Marker Gene Expression in all 38 Hematopoietic Cell Types
Rows and columns were clustered using average linkage with Euclidean distance. Each column represents the average expression of a gene over all its probe sets. CD45 and its isophorm CD45RA are represented by one column as well as IL-3RA and CD123.

Many of the marker genes are highly expressed in specific groups of cells. For example, CD71 is highly expressed in the erythrocytes but less expressed in the early stem and progenitor cells, megakaryocytes and B-cells and absent in all other cell types. Similarly, CD8 is expressed in only one subgroup of the T-cells namely T1-4. The early stem and progenitor cells and early erythrocytes also stand out with higher expression of CD34 and CD133. However, other marker genes that are listed beside specific cell types in the tree of Figure 1 are also present in other parts of the tree where they are not listed. For example, CD27 is used in Figure 1 to identify B-cells although it is also expressed in T-cells. Similarly, CD3 which is used to distinguish between N-cells is also present in T-cells and granulocytes. CD62L, a gene listed beside the T-cells in Figure 1 and is highly expressed in T2, T4, T6 and T8 but not expressed in T1, T3 and T7, is also present in N2 and N3 but absent in N1 and N4 and therefore can be used to distinguish between N-cells as well. CD62L is also present in the HSCs, monocytes and granulocytes and therefore is not a gene that characterizes T-cells only. CD22, which is listed in Figure 1 as a marker gene for the

basophil (BAS), is also present in the eosinophil (EOS), the B-cells and the dendritic cell (PLD). On the other hand, CD19, which is listed as a marker gene for the B-cells, is also moderately present in PLD, EOS and BAS. CD45, which is highlighted as either expressed or not expressed in many cell types including the T-cells, granulocytes and progenitor cells, happens to be highly present in the B-cells as well, although not listed beside that group of cells in Figure 1. Therefore, we conclude that although marker genes are useful in discriminating between different groups of cells in the differentiation tree as well as between different cell types within one group, many of them are not characteristic of one branch or one part of the tree but are expressed in different parts. In the next section, we show that many other genes, besides marker genes, are able to discriminate between distinct branches of the tree.

3.1 Thousands of Genes Discriminate Different Cell Lineages

The dendrogram of the 38 cell types shown on the left side of the heatmap in Figure 3, which was constructed via hierarchical clustering on the mean expression of the 27 marker genes, shows that some similar-type cells can be separated from other cell types by looking at marker gene expression alone. For example, the T-cells formed their own cluster, so did the B-cells, and the late erythrocytes E1-4. The early stem cells HS1-2 and progenitor cell MEP also formed one cluster with the early erythrocyte E1. On the other hand, there are other cells of similar types that were not grouped into one cluster such as the N-cells that formed a larger cluster with the late granulocytes G2-3 and monocytes MN1-2. However, the eosinohil EOS and basophil BAS were not part of this cluster but part of the larger cluster of megakaryocytes MG1-2 and early progenitors CMP and GMP. Therefore, although marker genes are able to discriminate a few groups of cells from others, we cannot completely rely on them as they fail to do so for other groups. Moreover, as their name indicates, marker genes code for specific proteins whose presence or absence in a cell is simply seen as a marker of the type of that cell. Those proteins may have no role in regulating cell differentiation or any biological functions that are active within a specific lineage of cells. Therefore there may be many other genes whose expression can offer a better description or indicator of the structure of the differentiation hierarchy. Unlike marker genes, those other

genes may exhibit functions that are more relevant to the structure of the tree such as metabolic functions, immunity response, cell signalling, transcriptional regulatory etc. Genes that are either up-regulated or down-regulated in one lineage alone are potentially involved in biological functions and processes that are unique to that lineage. On the other hand, genes that are up-regulated or down-regulated in more than one lineage are potentially involved in biological processes active within multiple lineages. In order to capture those genes and see how their expression relates to the differentiation tree, we define a parametric model (Model 1) that describes the gene expression pattern of an up-regulated and down-regulated gene along a lineage. In this model, shown in Figure 4, we define a lineage to be a vertical line of cells in the differentiation tree that starts with the root HS1 and ends with a fully-differentiated cell type. We therefore have 20 lineages in the tree of Figure 1. In cases when statistical significance of differential expression is not considered, we denote by e_i the expression level of a gene averaged over all its replicates in the i^{th} cell of a lineage of length n. However, when the statistical significance of differential expression is to be taken into account, e_i represents all expression values of a gene in all replicates. Therefore, the first point e_1 in Figure 4 represents the expression of a gene in the root HS1 at the top of the lineage, and the last point e_n represents the expression of a gene in the leaf-cell at the bottom of the lineage.

Model 1:

1- Up-regulated gene: A gene is up-regulated in a lineage if it shows a statistically significant increase of magnitude M between some e_i and e_{i+1} such that $e_j < e_i + \alpha M$ for all e_j where $j < i$ and $e_j > e_{i+1} - \alpha M$ for all e_j where $j > i + 1$, where $0 < \alpha < 0.5$.

2- Down-regulated gene: A gene is down-regulated in a lineage if it shows a statistically significant decrease of magnitude M between some e_i and e_{i+1} such that $e_j > e_i - \alpha M$ for all e_j where $j < i$ and $e_j < e_{i+1} + \alpha M$ for all e_j where $j > i + 1$, where $0 < \alpha < 0.5$.

3- Fluctuating gene: A gene is fluctuating if it shows at least one statistically significant increase or decrease between some e_i and e_{i+1}, however, no increase in expression exists between any e_i and e_{i+1} such that $e_j < e_i + \alpha M$ for all $j < i$ and $e_j > e_{i+1} - \alpha M$ for all $j > i + 1$, and no decrease in expression exists between any e_i and e_{i+1} such that $e_j > e_i - \alpha M$ for all $j < i$ and $e_j < e_{i+1} + \alpha M$ for all $j > i + 1$. Therefore, we consider a gene to be fluctuating if it shows a difference in expression between at least two successive cell types but still cannot be fit into the models of up-regulated and down-regulated genes.

4- Stable gene: A gene is stable if it does not show any significant increase or decrease between any two e_i and e_{i+1} and therefore is neither up-regulated nor down-regulated nor fluctuating.

Statistical significance of M is measured using a t-test with a p-value of 0.001. However, no statistical significance test is applied to the four inequality constraints $e_j < e_i + \alpha M$, $e_j > e_{i+1} - \alpha M$, $e_j > e_i - \alpha M$ and $e_j < e_{i+1} + \alpha M$.

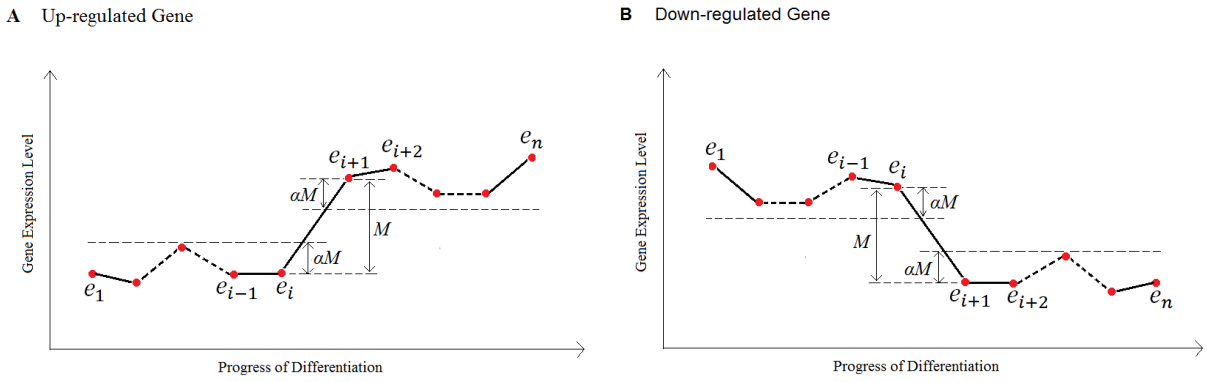


Figure 4 Model 1 for Up-regulated and Down-regulated Genes
A horizontal model (Model 1) of a gene's expression levels in a lineage of *n* cells. A lineage is a vertical line of cells in the differentiation tree that starts with the root HS1 and ends with one of the 20 fully-differentiated cell types. Point e_1 represents the expression level of the gene in the first cell in the lineage (the root HS1), and each other point e_i represents the expression level of the gene in the i^{th} cell in the lineage ($0 < \alpha < 0.5$ and $M > 0$).
(A) Up-regulated gene: $e_j < e_i + \alpha M$ for all $j < i$ and $e_j > e_{i+1} - \alpha M$ for all $j > i + 1$
(B) Down-regulated gene: $e_j > e_i - \alpha M$ for all $j < i$ and $e_j < e_{i+1} + \alpha M$ for all $j > i + 1$

With $\alpha = 0.5$, Model 1 labelled 13,041 genes as up-regulated in at least one lineage. Genes up-regulated in 14 or fewer lineages constituted 99.11% (12,925 genes) of the 13,041 genes and those up-regulated in only one lineage constituted 26.72% (3,484 genes). The model also labelled 12,384 genes as down-regulated in at least one lineage. However the number of genes down-regulated in 6 or fewer lineages was always lower than the number of up-regulated genes. Also, the number of genes down-regulated in 7 or more lineages was always higher than the number of up-regulated genes (Figure 5). Genes down-regulated in 14 or fewer lineages constituted 96.5% (11,951 genes) of the 12,384 genes and those down-regulated in only one lineage constituted 20.87% (2,584 genes). The model also labelled 778 genes as stable in all lineages and 4,369 not stable in any lineage. We then focused our attention on individual lineages and computed the mean number of up-regulated and down-regulated genes in each lineage for six values of α: 0, 0.1, 0.2, 0.3, 0.4 and 0.5. With the exception of the lineages of PLD, E5 and the B-cells, the mean number of down-regulated genes was larger than, in some cases double, the mean number of up-regulated genes in all other lineages (Figure 6).

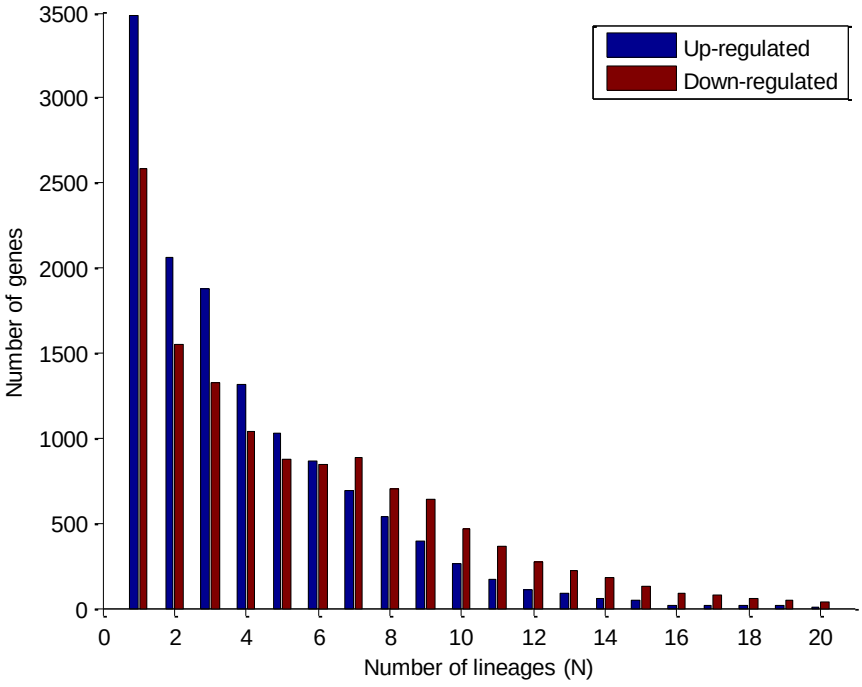


Figure 5 Number of Down-regulated and Up-regulated Genes in N Lineages by Model 1 ($\alpha = 0.5, p = 0.001$).

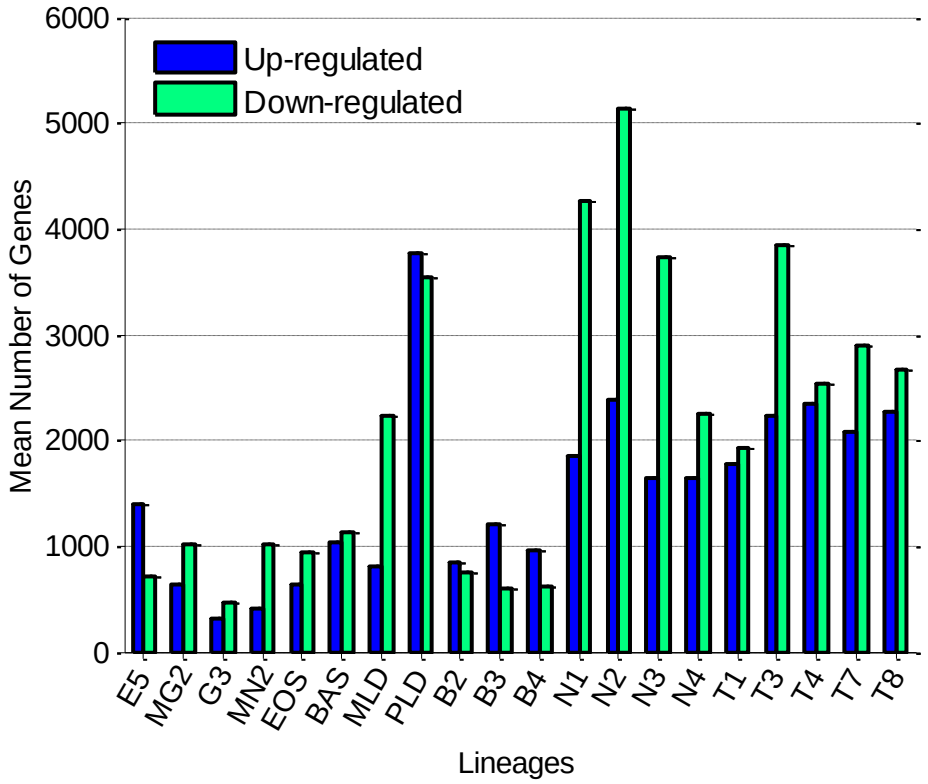


Figure 6 Mean Number of Down-regulated and Up-regulated Genes by Model 1
Each bar represents the mean number of genes in one lineage computed for 6 values of α in the range (0, 0.5) with an interval of 0.1 ($p = 0.001$). Labels on the x-axis represent the leaf cells in each lineage.

Once we knew there are thousands of genes that are either up-regulated or down-regulated in one or more branches of the tree, we were interested in seeing whether those genes are able to discriminate between those branches. We therefore assigned each lineage represented by its leaf node a new feature vector that spans only the 13,041 genes labelled up-regulated in at least one lineage. However, these new feature vectors are not gene expression vectors anymore but rather discrete vectors where a feature can hold one of the four distinct values {1, 2, 3, 4} depending on whether it is up-regulated (1), down-regulated (2), stable (3) or fluctuating (4) in a lineage. We then applied hierarchical clustering using hamming distance to the 20 new feature vectors such that the similarity between any two vectors is equal to the ratio of the number of features with equal values in both vectors to the total number of features (22215). Any code of four distinct numbers other than {1, 2, 3, 4} would still give

the same results with hierarchical clustering as long as hamming distance is used to measure similarity.

The clustering results in Figure 7 show that these genes strongly discriminate between different branches, and different cell types, in the differentiation tree. All five T-cell lineages formed one separate cluster and so did the four N-cell lineages which also formed a larger cluster with the PLD that shares with them the same ancestral line. The B-cell lineages grouped together and so did the EOS, BAS and MN2 lineages. However, the MG2 branch appears one cluster level away from its neighbouring E5 branch in the differentiation tree. This small discrepancy, together with the MLD lineage, resulted in the G3 branch being two cluster levels away from the other granulocytes. We then clustered the lineages using hamming distance on another set of feature vectors using the same mapping code {1, 2, 3, 4} but this time spanning the 12,384 genes that were labelled down-regulated in at least one lineage. The resulting tree we show on the left side of the heatmap in Figure 8 is exactly similar to the one in Figure 7. We therefore conclude that besides the marker genes highlighted in Figure 1, there are thousands of other genes that can be used to discriminate different branches in the tree when nonlinear expression rules (such as the ON/OFF rule) are used.

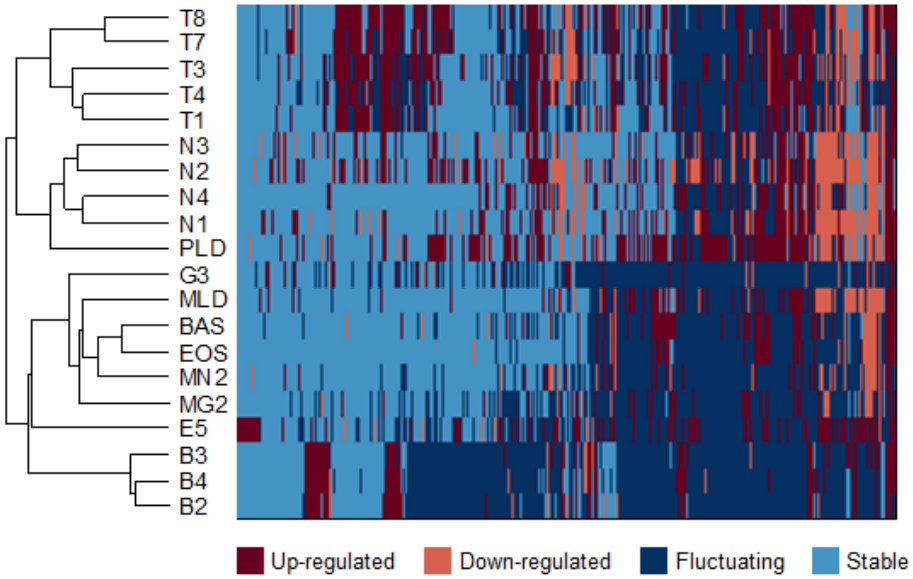


Figure 7 Heatmap of Up-regulated Genes Labelled by Model 1
The 13041 genes included in the heatmap are those that were labelled up-regulated in at least one lineage by Model 1 with $\alpha = 0.5$ and $p = 0.001$. Each column represents a gene and each row represents a lineage and is labelled by the leaf-cell of the lineage it represents. Rows and columns were clustered with average linkage using hamming distance. Heatmap was adjusted to screen resolution.

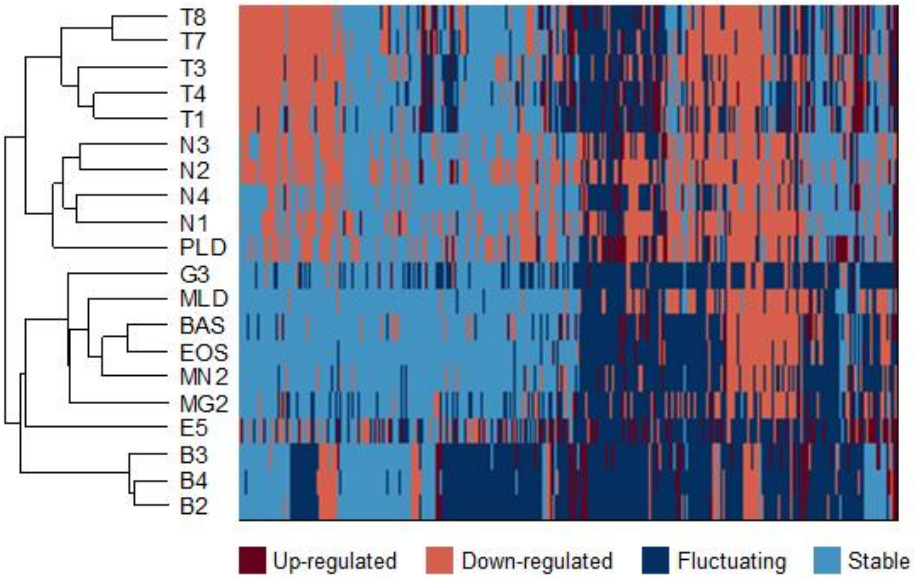


Figure 8 Heatmap of Down-regulated Genes Labelled by Model 1
The 12384 genes included in the heatmap are those that were labelled down-regulated in at least one lineage by Model 1 with $\alpha = 0.5$ and $p = 0.001$. Each column represents a gene and each row represents a lineage and is labelled by the leaf-cell of the lineage it represents. Rows and columns were clustered with average linkage using hamming distance. Heatmap was adjusted to screen resolution.

3.2 Applying Strict Statistical Significance Measures to Discriminating Genes

The results in the previous section clearly show that the selected genes are highly capable of discriminating different branches of the differentiation tree in Figure 1. However, the model we used to find those genes only considered the statistical significance of a gene's differential expression when searching for the two point e_i and e_{i+1} where the change in M occurs. We did not apply a statistical significance test when comparing a gene's expression levels at other points to the thresholds $e_i + \alpha M$ and $e_{i+1} - \alpha M$ for up-regulated genes and $e_i - \alpha M$ and $e_{i+1} + \alpha M$ for down-regulated genes. One may then ask whether some genes that were labelled either up-regulated or down-regulated would still be labelled so if the statistical significance test was applied to all points in the model when testing if they satisfy the inequality constraints. One may also ask whether more up-regulated and down-regulated genes would be detected if the two threshold lines located at a distance αM from the two points e_i and e_{i+1} were replaced by one line that can move freely between the two points. To answer these two questions we propose a second model (Model 2) that uses the same definition of a lineage as in Model 1 but enforces strict statistical significance requirements and also closes the gap between the two threshold lines. However, in this new model shown in Figure 9, each point e_i represents all expression values of a gene in all replicates of the i^{th} cell in the lineage and a t-test with a p-value of 0.001 is always used to measure the statistical significance of a gene's differential expression between any two cell types.

Model 2:

1- Up-regulated gene: A gene is up-regulated in a lineage if there exists an i where $1 < i \leq n$ such that $e_j < e_k$ for every j and k where $j < i \leq k$.

2- Down-regulated gene: A gene is down-regulated in a lineage if there exists an i where $1 < i \leq n$ such that $e_j > e_k$ for every j and k where $j < i \leq k$.

3- Fluctuating gene: A gene is fluctuating if it shows at least one statistically significant increase or decrease between some e_i and e_j, however, no i exists where $1 < i \leq n$ such that $e_j < e_k$ for every $j < i \leq k$ or $e_j > e_k$ for every $j < i \leq k$. Therefore, we consider a gene to be fluctuating if it shows a difference in expression between at least two cell types in the lineage but still cannot be fit into the models of up-regulated and down-regulated genes.

4- Stable gene: A gene is stable if it does not show any significant increase or decrease between any two e_i and e_j and therefore it is neither up-regulated nor down-regulated nor fluctuating.

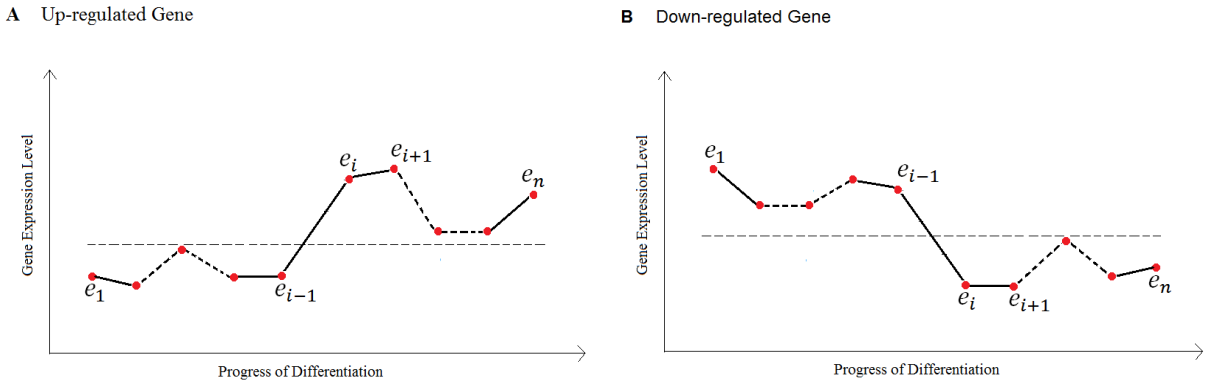


Figure 9 Model 2 for Up-regulated and Down-regulated Genes
A horizontal model (Model 2) of a gene's expression levels in a lineage of *n* cells. A lineage is a vertical line of cells in the differentiation tree that starts with the root HS1 and ends with one of the 20 fully-differentiated cell types. Point e_1 represents the expression level of the gene in the first cell in the lineage (the root HS1), and each other point e_i represents the expression level of the gene in the i^{th} cell in the lineage
(A) Up-regulated gene: $e_j < e_k$ for every j and k where $j < i \leq k$
(B) Down-regulated gene: $e_j > e_k$ for every j and k where $j < i \leq k$
Statistical significance of for each (j, k) pair is measured using a t-test.

With Model 2, 10,397 genes were labelled up-regulated in at least one lineage. Genes up-regulated in 14 or fewer lineages constituted 99.4% (10,335 genes) of the 10,397 genes and those up-regulated in only one lineage constituted 31.6% (3,285 genes). The model also labelled 10,699 genes as down-regulated in at least one lineage with the number of genes down-regulated in 4 or more lineages always exceeding the number of up-regulated genes (Figure 10). Genes down-regulated in 14 or fewer lineages constituted 98% (10,484 genes)

of the 10,699 genes and those down-regulated in only one lineage constituted 20.63% (2,207 genes). The model labelled only 35 genes as stable in all lineages and 5,255 not stable in any lineage. Similar to the case with Model 1, with the exception of the lineages of PLD, BAS, E5 and the B-cells, the number of down-regulated genes was larger than, in some cases double, the number of up-regulated genes in all other lineages (Figure 11).

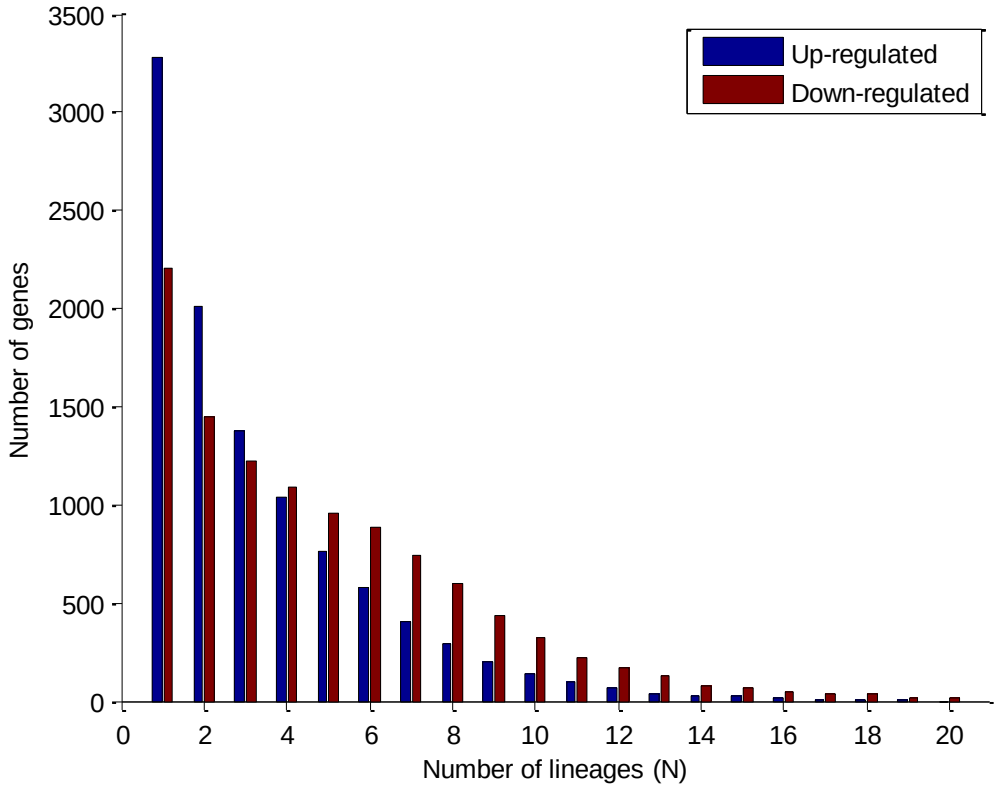


Figure 10 Number of Down-regulated and Up-regulated Genes in N Lineages by Model 2 ($p = 0.001$).

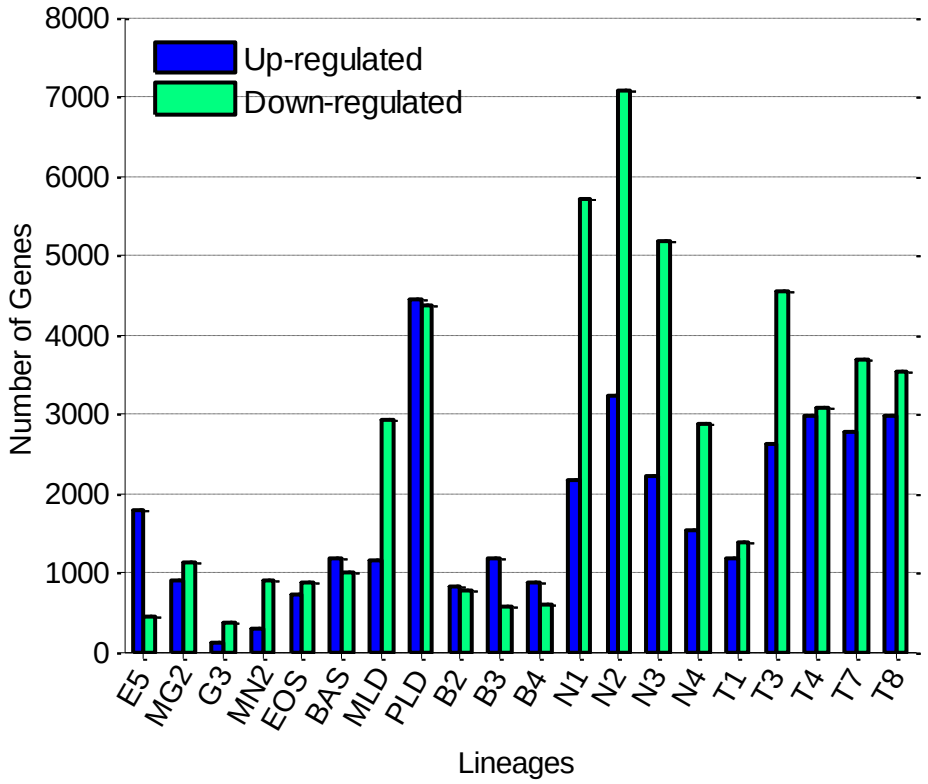


Figure 11 Number of Down-regulated and Up-regulated Genes by Model 2
Labels on the x-axis represent the leaf cells in each lineage ($p = 0.001$).

Again, we were interested in seeing whether these genes, with their new labels, are still able to discriminate different branches of the differentiation tree and if so are they any better than the labels we got from Model 1. Therefore, using the same mapping code {1, 2, 3, 4} we used with the labels of Model 1, we assigned each lineage a new feature vector that spans only the 10,397 genes labelled up-regulated by Model 2 in at least one lineage and we applied hierarchical clustering to those vectors using hamming distance and produced the heatmap shown in Figure 12. We then used the same mapping to assign each lineage a new feature vector that spans the 10,699 down-regulated genes and we applied hierarchical clustering to those vectors using hamming distance as well and produced the heatmap shown in Figure 13. Similar to the hierarchical trees we got from Model 1, the T-cell lineages formed their own cluster in both trees and so did the N-cell lineages and the B-cell lineages. However, this time both trees put the G3 branch and the MN2 branch, it's neighbour in the

original differentiation tree, in their own cluster, a result we were not able get with Model 1. Moreover, in the tree of Figure 13 which was constructed using down-regulated genes, the G3 and MN2 branches also clustered with the other two granulocyte branches, the BAS lineage and EOS lineage. This allowed the E5 lineage to cluster with its nearest neighbour branch in the differentiation tree, the MG2 lineage, a result that we were also not able to get with Model 1. Therefore, the adjustments we made in Model 2 lead to improvements in the trees produced by hierarchical clustering specifically in the clusters including the granulocytes, erythrocytes and megakaryocytes.

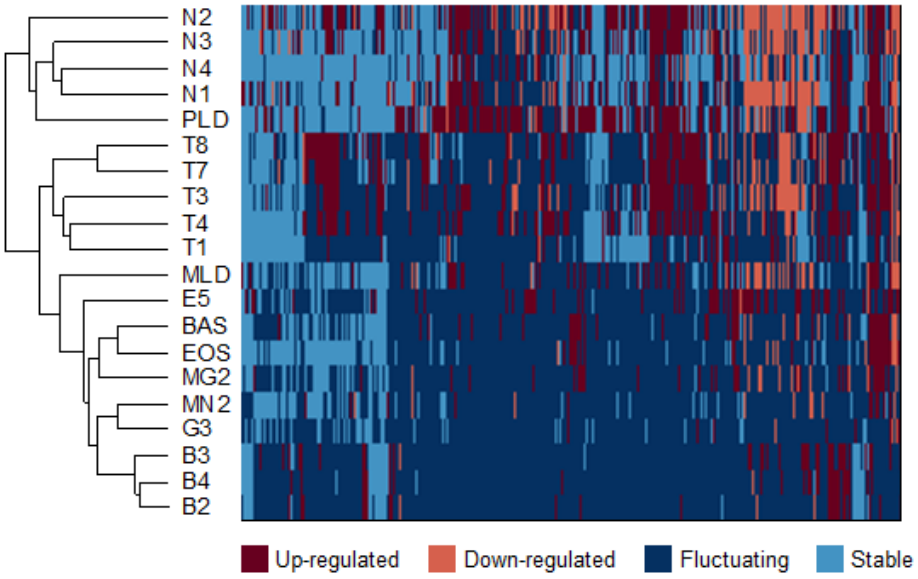


Figure 12 Heatmap of Up-regulated Genes Labelled by Model 2
The 10397 genes included in the heatmap are those that were labelled up-regulated in at least one lineage by Model 2 with $p = 0.001$. Each column represents a gene and each row represents a lineage and is labelled by the leaf-cell of the lineage it represents. Rows and columns were clustered with average linkage using hamming distance. Heatmap was adjusted to screen resolution.

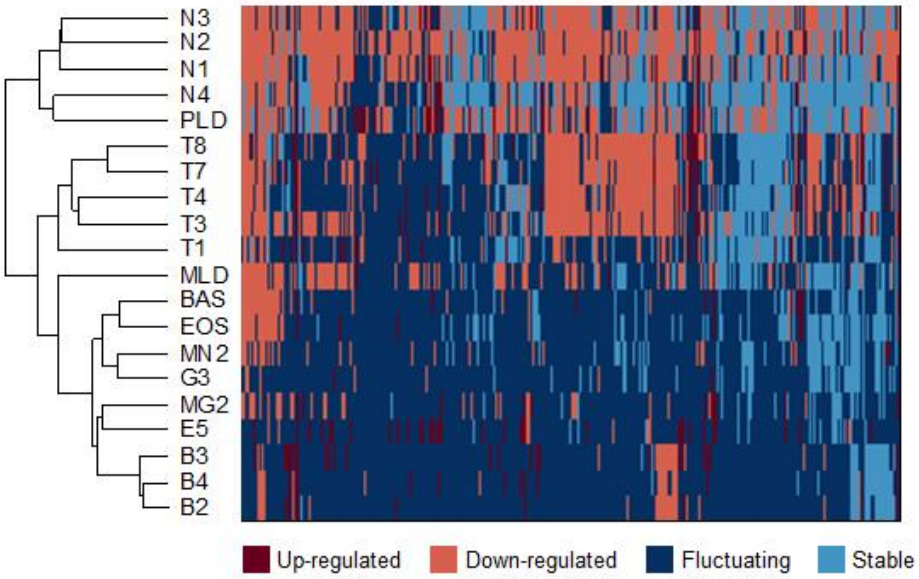


Figure 13 Heatmap of Down-regulated Genes Labelled by Model 2
The 10699 genes included in the heatmap are those that were labelled down-regulated in at least one lineage by Model 2 with $p = 0.001$. Each column represents a gene and each row represents a lineage and is labelled by the leaf-cell of the lineage it represents. Rows and columns were clustered with average linkage using hamming distance. Heatmap was adjusted to screen resolution.

Out of the 22215 genes, 13,232 were labelled up-regulated by at least one of the two models in at least one lineage and most of them are up-regulated in more than one lineage. However, we were interested in seeing whether those labels happen to be in the same lineages or in different lineages since that would give us a more accurate report on how often the two models agree on the labels assigned to each gene. Therefore, for each of those genes, we counted the number of lineages in which it was labelled up-regulated by either model and we found a total of 54,707 gene-lineage pairs. Out of these 54,707 pairs 31,495 were labelled by both models, 19,583 were labelled by Model 1 only and 3,629 by Model 2 only. Similarly, for the 12,680 genes labelled down-regulated by at least one model, we have a total of 69,263 gene-lineage pairs. Out of these 69,263 pairs 47,253 were labelled by both models, 18,273 were labelled by Model 1 only and 3,737 by Model 2 only. To get a better view of how often those labels occurred in the same lineages, we assigned a new feature vector to each lineage that spans the 13,232 up-regulated genes only where a feature can hold one of the four distinct values {1, 2, 3, 4} depending on whether it was labelled up-regulated in that lineage by both models (1), by neither model (2), by model 2 only (3), or by

model 1 only (4). We then applied hierarchical clustering using hamming distance to the 20 new feature vectors and produced the heatmap shown in Figure 14. We then repeated the same process on the 12,680 down-regulated genes and produced the heatmap shown in Figure 15.

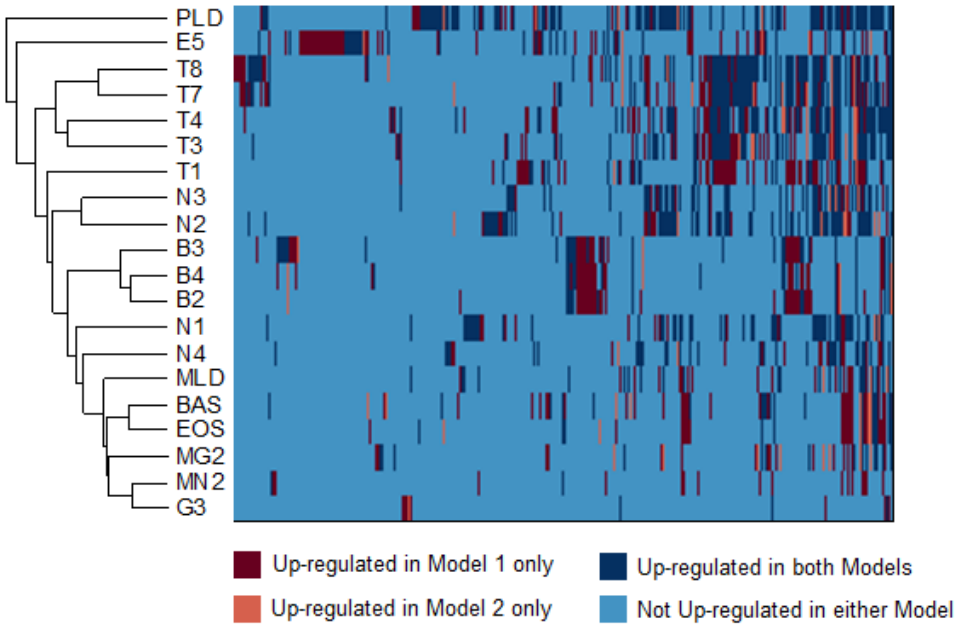


Figure 14 Heatmap of Up-regulated Genes Labelled by both Models
The 13232 genes included in the heatmap are those that were labelled up-regulated in at least one lineage by at least one of the two models with $\alpha = 0.5$ and $p = 0.001$. Each column represents a gene and each row represents a lineage and is labelled by the leaf-cell of the lineage it represents. Rows and columns were clustered with average linkage using hamming distance. Heatmap was adjusted to screen resolution.

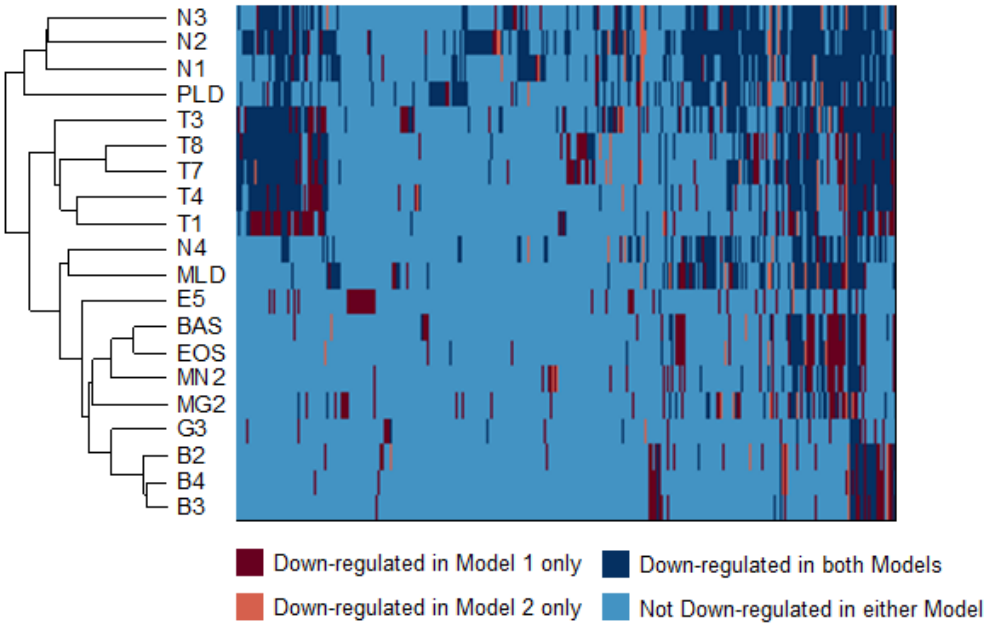


Figure 15 Heatmap of Down-regulated Genes Labelled by both Models
The 12680 genes included in the heatmap are those that were labelled down-regulated in at least one lineage by at least one of the two models with $\alpha = 0.5$ and $p = 0.001$. Each column represents a gene and each row represents a lineage and is labelled by the leaf-cell of the lineage it represents. Rows and columns were clustered with average linkage using hamming distance. Heatmap was adjusted to screen resolution.

The heatmaps in Figures 14 and 15 show that the majority of genes labelled up-regulated or down-regulated by both models were assigned those labels in the same lineages. It is also clear that many genes that were detected by Model 1 were dismissed by Model 2 in some lineages due to the statistical significance constraints applied in the latter model. This disagreement between the two models also reflected on the dendrograms shown on the left side of both heatmaps which do not show consistency with the true differentiation tree as much as the previous dendrograms we produced have shown. However, we can still say that both models were highly successful in providing gene labels that discriminated between different branches of the tree although Model 2 proved to be more accurate and reliable. In the presence of all these promising genes, it remains to be seen whether any existing tree construction methods are capable of reconstructing the differentiation hierarchy of the 38 cell types from their gene expression data.

3.3 Hierarchical Clustering does not Produce the Proper Type of Tree

Kluger et al. (2004) showed that single linkage hierarchical clustering can classify cells types, specifically human blood cells, into nine distinct lineages. However, their data consisted of gene expression data for fully-differentiated cell types only. Therefore, it was not clear from their work how well hierarchical clustering can perform on gene expression data of cell types at different stages of differentiation and not only at the fully-differentiated level. We were sure, though, that hierarchical clustering cannot perfectly reproduce the differentiation tree of Figure 1 because nodes clustered by this technique always end up being leaves that are connected through intermediate nodes created by the clustering method itself. Every time hierarchical clustering connects a new cell to a cell or cluster already in the tree, it needs to create a new intermediate node, thus pushing other cells that will later be added further away from that cluster. These intermediate nodes create a large structural difference between the constructed tree and the true differentiation tree. To better illustrate the pros and cons of hierarchical clustering we apply it to the expression data we have for all 38 cell types using the three linkage methods: single, average and complete linkage (see descriptions in Chapter 2) and the five distance metrics described in Chapter 2: Euclidean, L1, cosine, correlation and Chebychev.

In Figure 16, we show the hierarchical clustering tree constructed using average linkage with Euclidean distance. We choose to show this tree specifically because, as we show later, it accumulates the lowest error among all hierarchical clustering trees we constructed. Similar to the other trees, this tree succeeds in grouping some cells of similar types together but also fails with other cell types. For example, all T-cells are in one cluster but they are not organized the same way they are in the differentiation tree. The B-cells also formed one cluster and so did the late erythrocytes E3, E4 and E5. However, the other two erythrocytes E1 and E2 clustered with their progenitor MEP and the two early stem cells HSC1 and HSC2. This association between the early erythrocytes and their progenitor is also evident in the results of Novershtern et al. (2011). Two of the N-cells, N2 and N3, clustered with the T-cells while N1 clustered with the monocyte MN1 and dendritic cell MLD and N4 clustered with CMP, PLD and the two pre-B cells. So, it seems that this hierarchical clustering tree did

a poor job at grouping the N-cells in one cluster which we would have expected to be close to the T-cell cluster. Similarly, the monocytes and granulocytes which are all descendents of the progenitor GMP are not all in one cluster but are distributed among different clusters that include cells of different types. The BAS and EOS cells are clustered with the megakaryocytes MG1-2. G1 is clustered with the B-cells, and the early monocyte MN1 is clustered with N1 and PLD while the other more differentiated cell types G2, G3 and MN2 clustered together with the stem cells and early erythrocytes.

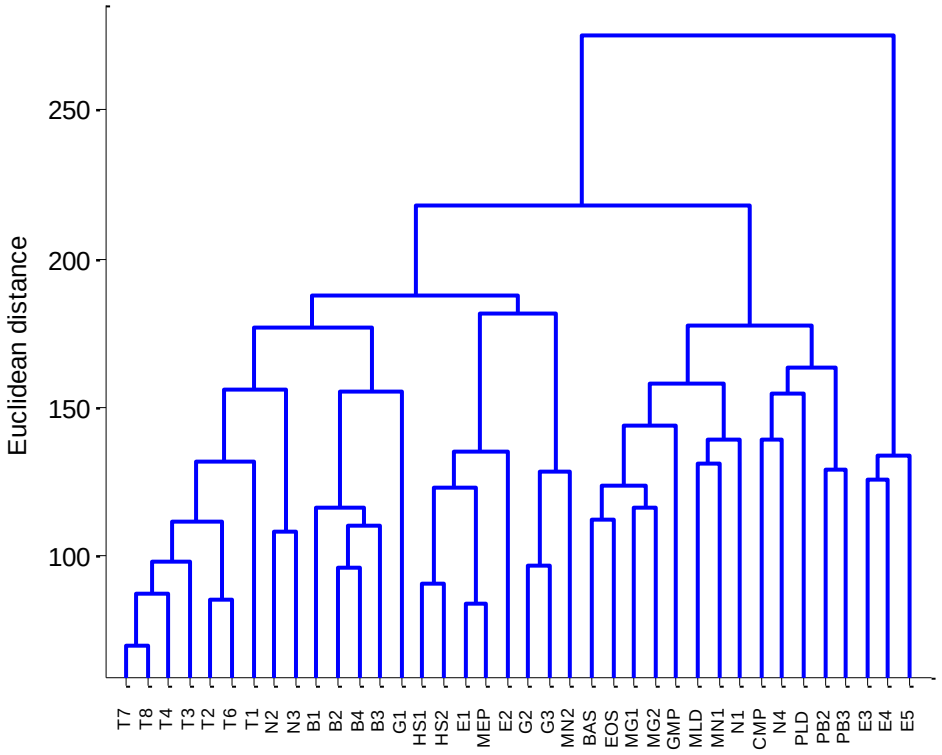


Figure 16 Hierarchical Clustering Tree for all 38 Cell Types
Cell types clustered using average linkage with Euclidean distance on all 22215 gene expression variables in the Novershtern data.

The other trees we constructed with hierarchical clustering also clustered the cell types with varying levels of accuracy. To better quantify the discrepancies in these trees, we measure the error in each tree by considering how far each node in the tree is from its true parent. More precisely, for each node and its parent in the differentiation tree of Figure 1, the error for that particular node in another constructed tree T is equal to the number of nodes located

on the path connecting it to that parent in T. For a hierarchical clustering tree, the error is calculated for leaf nodes only since all intermediate nodes do not exist in the original differentiation tree and thus have no true parent. We then define the total error in T as the sum of errors from all its nodes and the mean error in T as the mean of all node errors. In Table 2 we show the errors for all hierarchical clustering trees constructed using different linkage methods with all five distance metrics.

		Euclidean	L1	Cosine	Correlation	Chebychev
Single Linkage	Total	227	212	264	270	352
	Mean	6.135	5.730	7.135	7.297	9.514
Average Linkage	Total	202	203	244	266	249
	Mean	5.460	5.487	6.595	7.189	6.730
Complete Linkage	Total	233	238	231	254	257
	Mean	6.297	6.432	6.243	6.865	6.946

Table 2 Errors in Hierarchical Clustering Trees
Total and mean error in hierarchical clustering trees constructed using different distance metrics and linkage methods on expression data of the 38 hematopoietic cell types. Mean error was calculated by dividing the total tree error by 37, the number of all cell types excluding the root HSC1.

Generally, trees constructed using Euclidean and L1 distance metrics have less error than trees constructed using the other three distance metrics. With Euclidean and L1 metrics, trees constructed using average linkage had less error than trees constructed using single and complete linkage. With cosine and correlation metrics, complete linkage produced a tree with less error than the other linkage methods. On the other hand, it produced a tree with the largest error when the Chebychev metric was used. However, regardless of the amount of error in each tree and how we choose to measure it, any tree constructed by hierarchical clustering is not satisfactory because it is not the proper type of tree we aim to produce. Nevertheless, it was still interesting to us to see whether hierarchical clustering would perform any better if we were to throw out expression data of all intermediate cell types and only keep the data for fully-differentiated cell types. We therefore applied hierarchical clustering to the 20 mature cell types in the Novershtern data using average linkage with the following four distance metrics: Euclidean, L1, cosine, and correlation. Although all four trees, shown in Figure 17, successfully grouped all T-cells in one cluster and all B-cells in one cluster, they all fail to group the N-cells in one cluster and also fail to group the E5 with

MG2 which is the leaf-cell of the branch nearest to the erythrocyte branch. The four trees also fail to group BAS and EOS with G3 and MN2 which share the ancestor GMP in the differentiation tree. Therefore, besides the fact that we had to throw out nearly half of the expression data, hierarchical clustering still fails to properly discriminate between different branches of the tree even when we spare it from having to deal with intermediate cell types.

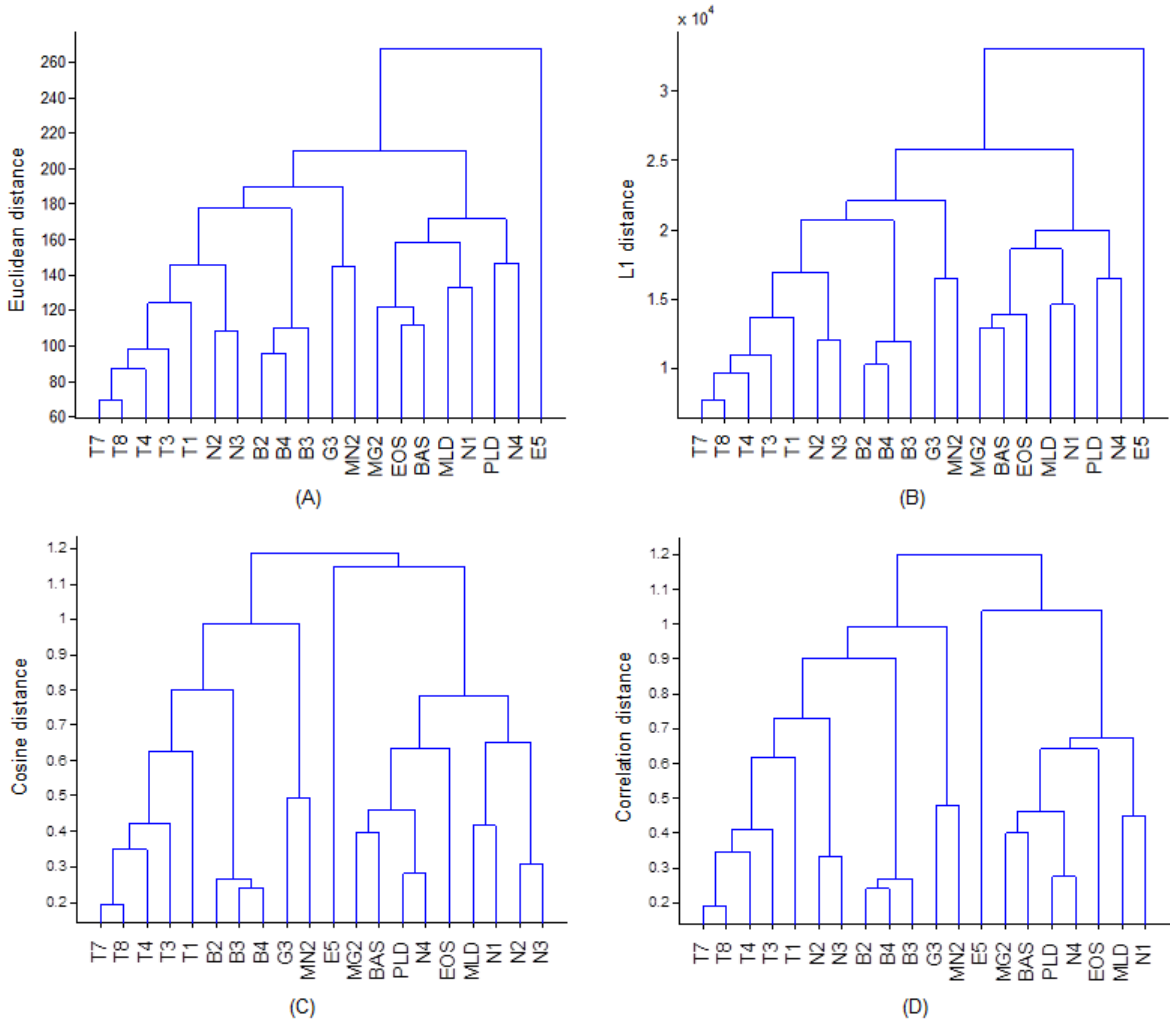


Figure 17 Hierarchical Clustering Trees for the 20 Fully-differentiated Cell Types
Cell types clustered using average linkage on all 22215 genes in the Novershtern data. Distance metrics used were (A) Euclidean distance, (B) L1 norm, (C) cosine distance, and (D) correlation distance.

3.4 Parsimonious Maximization also does not Produce the Proper Type of Tree

Joshi et al. (2011) used the PARS program of the PHYLIP package (Felsenstein 1981) to show that, similar to phylogenetics, the differentiation hierarchies of mammalian tissue - specifically hematopoietic differentiation, neural differentiation and early endoderm organogenesis - can be inferred by parsimonious maximization. They digitized their gene expression data using two different methods into 2 states, 0 and 1, where 0 stands for a gene that is not expressed in a cell and 1 stands for an expressed gene. For the first digitization method, a gene expression value was set to 1 if it was greater than a constant cut-off value. They used 5 cut-off values: 50, 100, 150, 200 and 250. For the second digitization method, a gene expression value was set to 1 if it was larger than X% of the highest value. They also used 5 values for X: 30, 40, 50, 60 and 70. Their results show that the tree predicted by the PARS program was largely unaffected by the digitization method or cut-off value used. However, although their results do show consistency between the three parsimonious trees and the current experimentally validated trees, again they only utilized gene expression data at the fully-differentiated level. They later predicted the expression states of intermediate nodes as well as some transcriptional programmes from the digitized data of the leaf-nodes present in the predicted tree.

Similar to hierarchical clustering, we did not expect parsimonious maximization to be able to accurately reconstruct a differentiation hierarchy when gene expression data at the intermediate levels is used because it also produces a tree where all nodes are leaves. Although this method does not put nodes into clusters the way hierarchical clustering does, it follows a branching technique that ends up creating branching points that are more like intermediate nodes. Therefore, the tree resulting from maximization of parsimony looks more like a hierarchical clustering tree with nodes branching off at different stages. Nevertheless, we were interested in seeing whether the data digitization required by this method using different cut-off values would provide any improvements in grouping the 38 hematopoietic cell types over hierarchical clustering. Therefore, we applied it to the Novershtern data after digitizing each gene expression value to either 0 or 1 using the X%

cut-off method that was used in Joshi et al. (2011). However, since we had normalized the expression of each gene so that its mean across all cell types is zero we cannot just use an X% of the maximum value as a cut-off because then all expression values less than zero would always be mapped to 0 regardless of the cut-off value used. Therefore, we used a cut-off value equal to the minimum value plus X% of the difference between the minimum and maximum values. We then constructed five trees using the PARS program with seven different values of X: 20, 30, 40, 50, 60, 70 and 80. We then calculated the errors shown in Table 3 for the seven trees the same way we did for the hierarchical clustering trees.

X		20 %	30 %	40 %	50 %	60 %	70 %	80 %
Error	Total	46.5	164.4	237	292.5	202	118.5	59
	Mean	1.257	4.443	6.405	7.905	5.459	3.203	1.595

Table 3 Errors in Parsimonious Trees
Total and mean error in parsimonious trees constructed from the expression data of the 38 hematopoietic cell types. Mean error was calculated by dividing the total tree error by 37, the number of all cell types excluding the root HSC1. Gene expression data was digitized to 0 or 1 using the cutoff: $min + \mathrm{X}(max - min)$

All values of X, except for X = 40 % and 60 %, produced multiple maximally parsimonious trees with equal total change in expression. However, the trees produced by the same value of X were always very similar thus their errors were very close, sometimes equal. Hence, the error for each value of X in Table 3 is the average of the errors for all trees that value produced. We realized that although the errors for X = 20 % and X = 80 % were very low, the trees produced by these values of X were very close to the simple tree in which all 38 cell types are connected to one parent node. For that simple tree, the error would be 37, the minimal error possible with the method of parsimonious maximization since each cell is one node away from its true parent. This result is expected when the cut-off value is either too low causing all expression data to be mapped to 1 or too high causing all data to be mapped to 0. In both cases, the pairwise distances between all cell types are equal to zero thus producing a tree with all cell types connected to each other through one and only one intermediate node. We therefore choose to show in Figure 18 one of the two parsimonious trees produced by the cut-off value X = 50 % which has the largest error but a structure that

is far from the simple structures of the trees produced using cut-off values closer to the extremes.

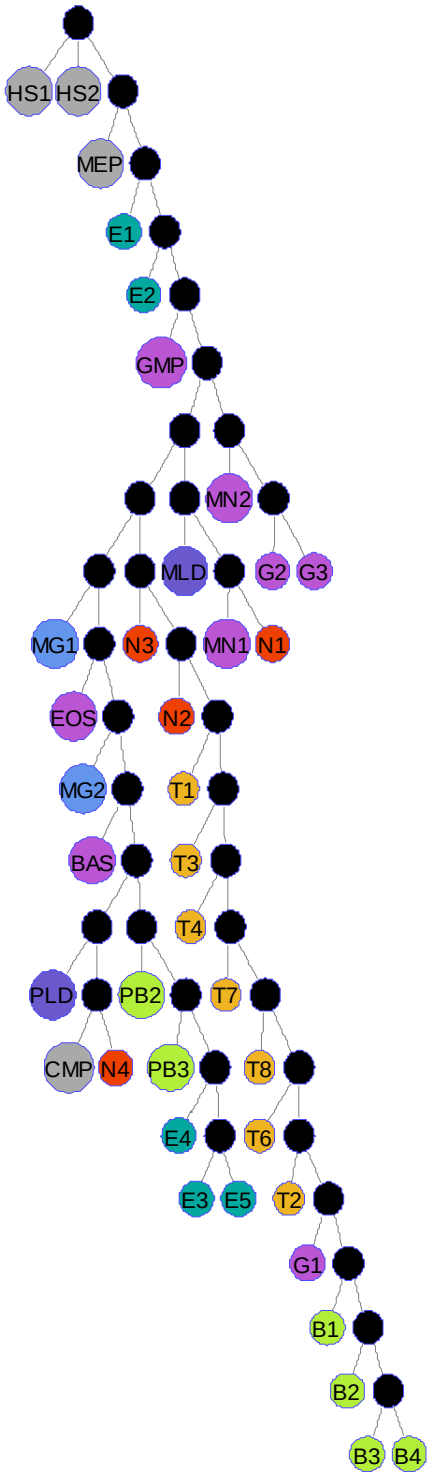


Figure 18 Maximally Parsimonious Tree for all 38 Cell Types
Expression of all 22215 genes was digitized into 2 states, 0 and 1, with a cut-off X = 50%.

Joshi et al. mentioned that the cut-off value used for digitization did not have much effect on the structure of the maximally parsimonious tree produced. However, that might be because their data consisted of fully-differentiated cell types only. Our results show that when expression data for intermediate-level cell types is also included, the cut-off value has a large effect on the structure of the maximally parsimonious tree. The closer the cut-off value is to the extremes 0 and 100% the closer the produced tree is to the simple tree where all cell types are children of one and only one node. Although cut-off values closer to the extremes produced trees with lower errors than most trees produced by hierarchical clustering, a cut-off of 50% produced a tree with larger error than most of the hierarchical clustering trees. Determining what cut-off value is best to use when digitizing the expression data is therefore a big challenge. Nevertheless, regardless of the cut-off value used, maximization of parsimony does not produce the type of tree we aim to construct and therefore is not a satisfactory option.

3.5 A Minimum Spanning Tree is the Proper Type of Tree

Unlike hierarchical clustering and parsimonious maximization, a minimum spanning tree (MST) produces the proper type of tree, a tree where nodes are not all classified as a leaves. Given a set of nodes *N* and a set of weighted edges *E* that connect each two nodes in *N* together, a minimum spanning tree is the tree that spans all nodes of *N* while minimizing the total sum of weights on all its edges. In our case, the weight of an edge is the distance between the two nodes connected by that edge calculated using any distance metric we choose. We therefore constructed MSTs on all 38 cell types in the Novershtern data using five distance metrics: Euclidean, L1, cosine, correlation and Chebychev. We then calculated the error in each tree same as we did with previous trees. The errors shown in Table 4 for all trees we obtained are lower than the errors in all trees we obtained by hierarchical clustering and also lower than the trees we obtained by parsimonious maximization using X = 40%, 50% and 60%. Although the errors for some of the MSTs are not as low as those of the parsimonious trees we obtained using X = 30%, 70% and 80%, the trees produced using these cut-off values are not reliable despite their low errors because they are very similar to

the minimal-error tree which we could have obtained by simply mapping all expression data to a value of 1.

		Euclidean	L1	Cosine	Correlation	Chebychev
All genes	Total	132	111	174	180	91
	Mean	3.568	3	4.703	4.865	2.459
Marker genes	Total	99	89	162	150	70
	Mean	2.676	2.405	4.378	4.054	1.892

Table 4 Errors in Minimum Spanning Trees
Total and mean error in minimum spanning trees of the 38 hematopoietic cell types constructed using different distance metrics from the expression data of both: all genes and marker genes only. Mean error was calculated by dividing the total tree error by 37, the number of all cell types excluding the root HSC1.

Using only marker gene expression to calculate distances between nodes always resulted in trees with less error compared to trees constructed using expression of all genes. The error was lowest with Chebychev distance regardless whether expression of all genes was used or that of marker genes only. Although, with all five distance metrics, cell types were not connected exactly the way they are connected in the differentiation tree of Figure 1, cells of similar types were usually located near each other. Looking at the MSTs in Figures 19, 20, 21, 22 and 23 we see that whenever marker genes were used to construct a MST, the erythrocytes formed a lineage of cells although usually not in the correct order. Although that lineage was missing E1 in the trees produced by Euclidean and Chebychev metrics, this result was not possible with hierarchical clustering and parsimonious maximization. Most importantly, unlike the previous methods we explored, no “stranger” nodes are added in a MST thus making it a plausible way to reconstruct the differentiation tree if we use the right distance metric.

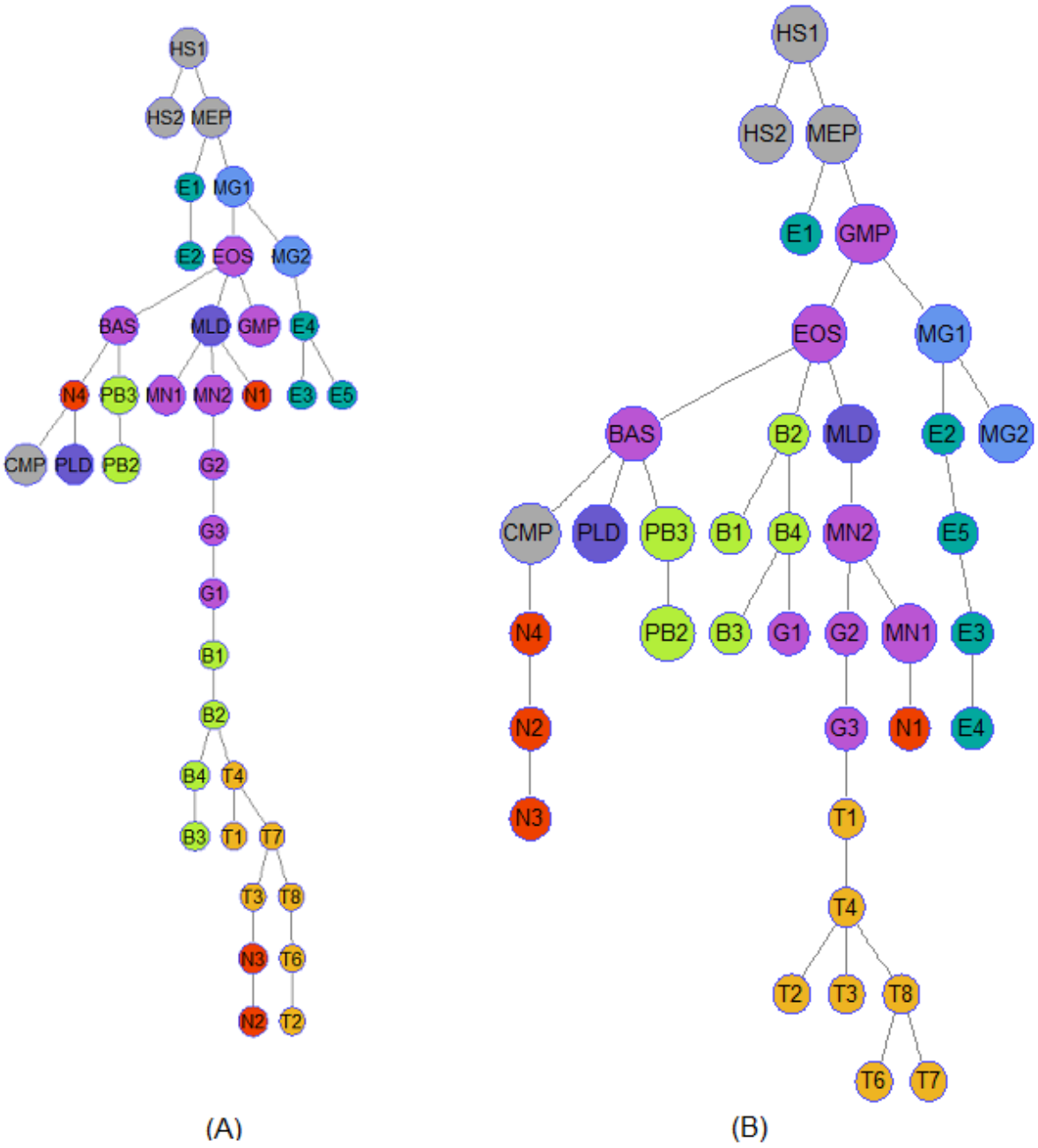


Figure 19 Minimum Spanning Trees using Euclidean Distance on all 38 Cell Types
Trees constructed on expression data of (A) all genes and (B) marker genes only.

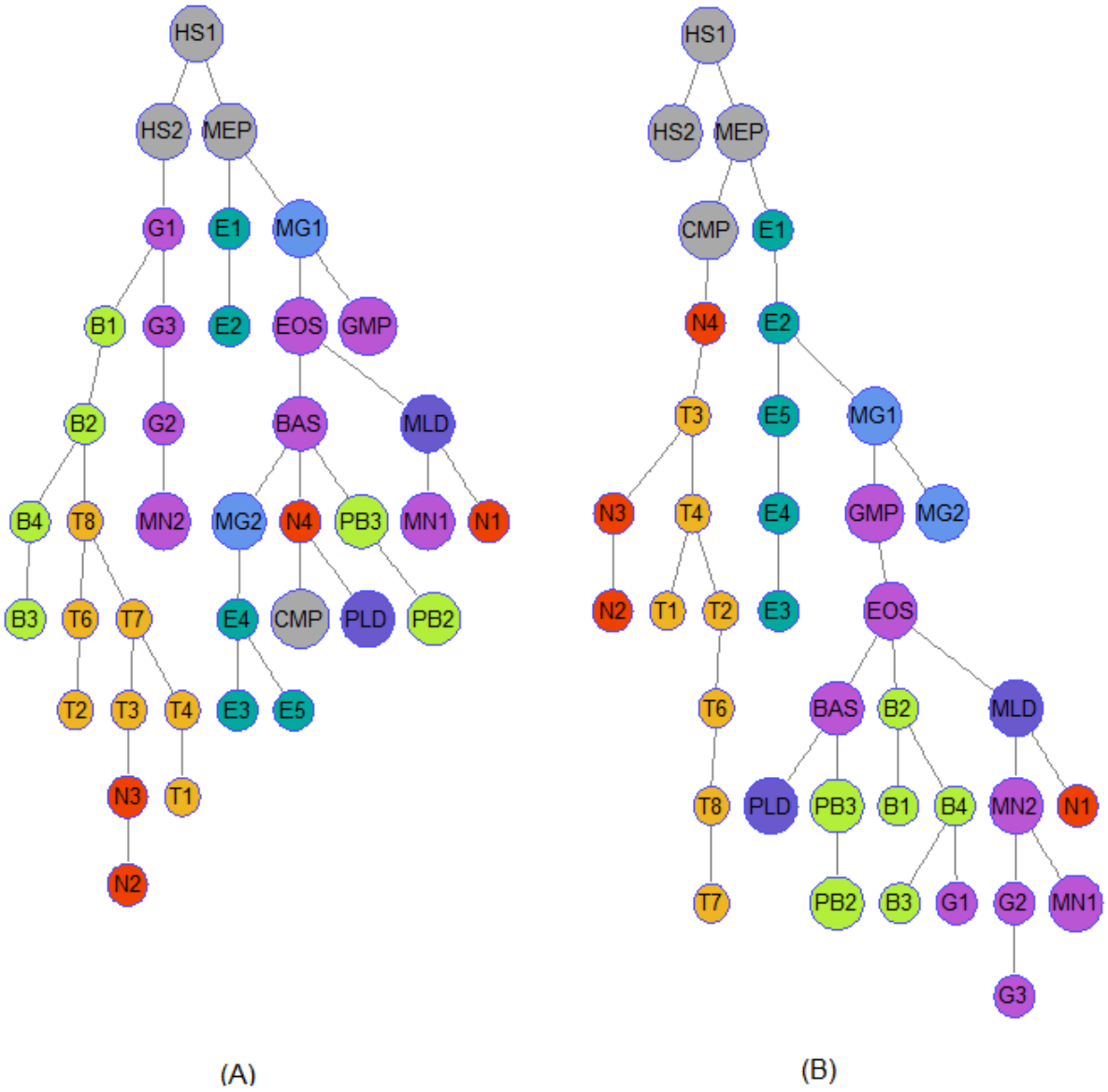


Figure 20 Minimum Spanning Trees using L1 Distance on all 38 Cell Types
Trees constructed on expression data of (A) all genes and (B) marker genes only.

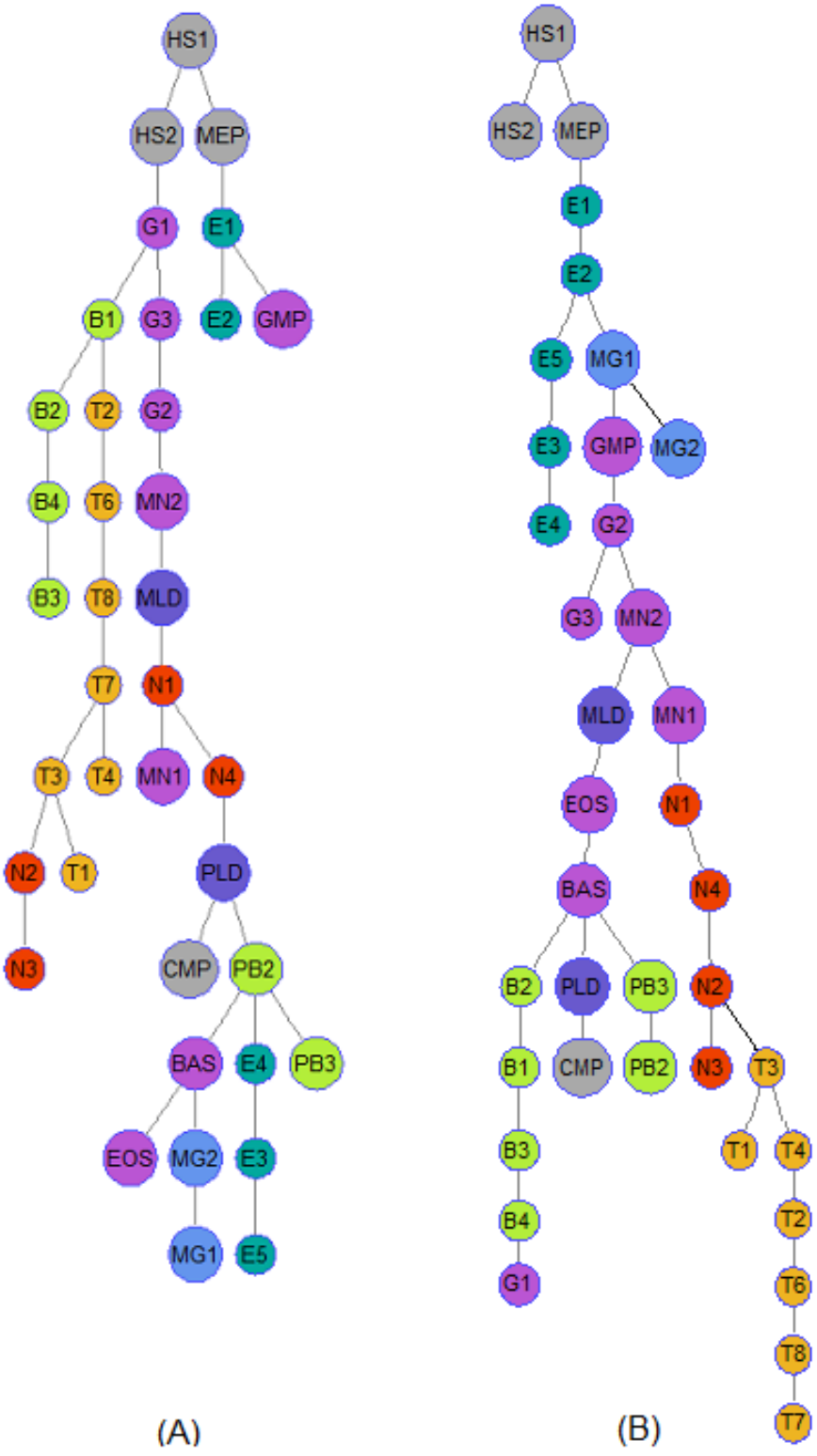


Figure 21 Minimum Spanning Trees using Cosine Distance on all 38 Cell Types
Trees constructed on expression data of (A) all genes and (B) marker genes only.

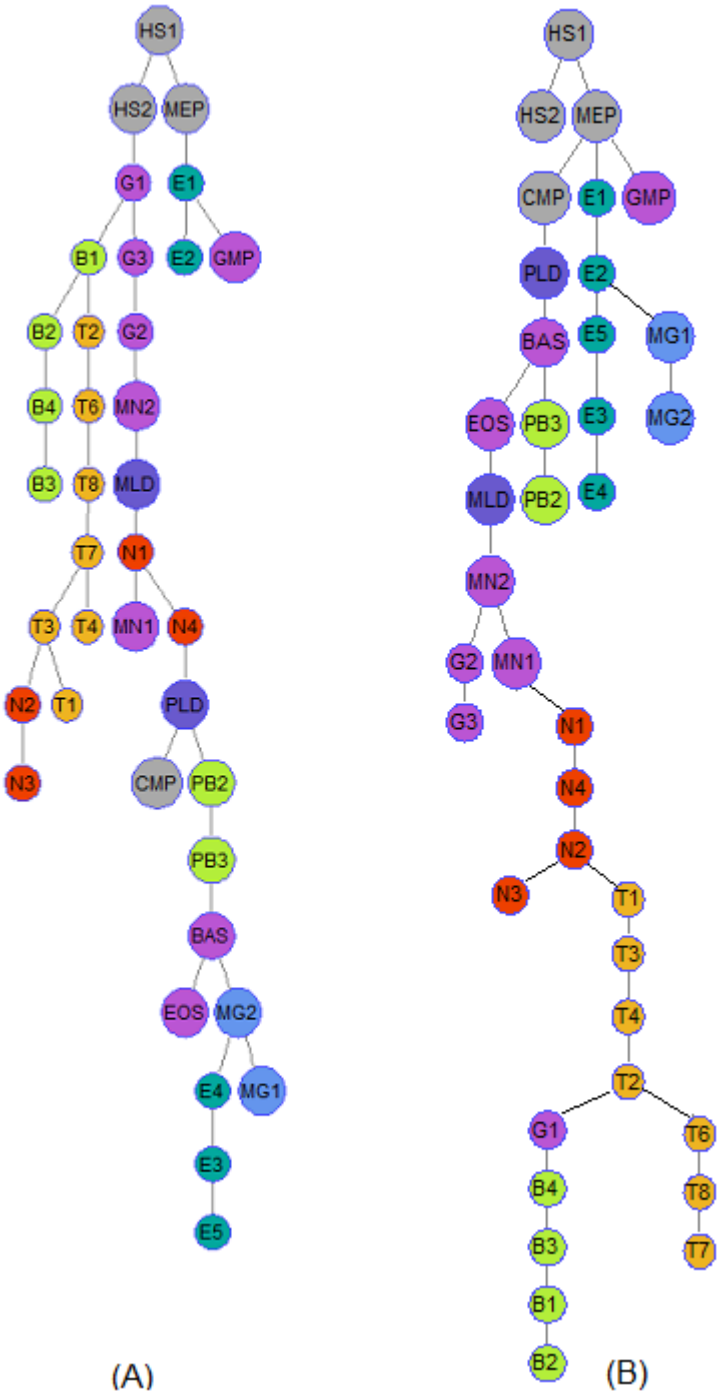


Figure 22 Minimum Spanning Trees using Correlation Distance on all 38 Cell Types
Trees constructed on expression data of (A) all genes and (B) marker genes only.

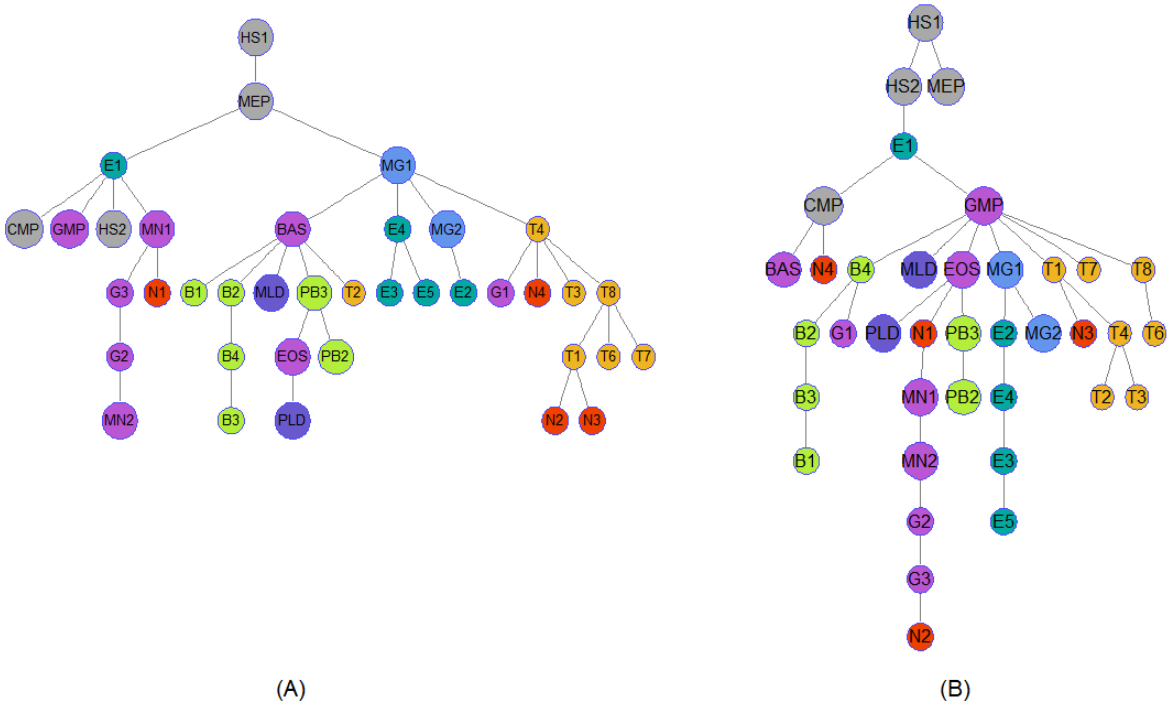


Figure 23 Minimum Spanning Trees using Chebychev Distance on all 38 Cell Types
Trees constructed on expression data of (A) all genes and (B) marker genes only.

In this chapter, we proposed two novel models to assign labels to genes based on their expression pattern in each lineage. We found thousands of genes whose assigned labels can discriminate between different branches in the hematopoiesis differentiation tree. However, even in the presence of all these important genes, hierarchical clustering and maximization of parsimony are not able to reconstruct the differentiation hierarchy because they were designed to construct trees of different forms, though hierarchical in nature. A minimum spanning tree, however, produces the proper type of tree but not the correct tree with the five standard distance metrics we used. In the next chapter, we define a set of constraints that are provably sufficient to reconstruct the differentiation tree in its correct form as a minimum spanning tree and we propose two approaches that aim to reconstruct the tree by satisfying these constraints. Each approach performs two tasks concurrently, the first task is to learn the correct distance metric that allows for tree reconstruction and the second task is to select the fewest number of genes to participate in that distance metric.

4 APPROACHES TO FINDING A WEIGHTED DISTANCE METRIC

In this chapter, we propose a novel solution to the following problem: Given a set of objects (cell types) with associated feature vectors (gene expression), and given a tree T on those objects, identify a weighted Euclidean distance metric such that the minimum spanning tree M constructed with that distance metric is precisely the tree T. We first propose a set of conditions on the pairwise distances between the nodes of T and prove that they are sufficient (though not all necessary) to produce a minimum spanning tree with the exact same structure of T. We then propose a mean square error minimization approach to finding a weighted Euclidean metric by assigning a fixed value to those pairwise distances such that those conditions are all met. However, since our choice of the pairwise distances is arbitrary, the MSE minimization is not guaranteed to find a weighted vector even if that vector does exist unless we happen to choose the correct pairwise distances. Therefore, we translate those conditions to a set of linear constraints on the weights, depending on the tree T and feature vectors, and propose an efficiently-solvable linear program designed to satisfy these constraints while minimizing the L1-norm of the weights. This produces a "sparse" set of weights for hierarchy reconstruction, in essence choosing a subset of genes to participate in the distance metric.

4.1 Pairwise Distances and Tree Reconstruction

Our central theoretical result establishes a set of sufficient conditions on pairwise distances between objects such that their minimal spanning tree has a given, desired structure.

Theorem 4.1: Given a rooted tree T with a set of nodes N and a pairwise symmetric distance matrix $D: N \times N \to R_0^+$ over the nodes of N, suppose that for any three nodes i, j and k in N where the path from node i to node k in T passes through node j the following is true

$$D(i,j) < D(i,k) \qquad (4.1)$$

Then T is a minimum spanning tree of the nodes N.

Proof: Without loss of generality we can imagine constructing a minimum spanning tree M based on distance matrix D using Prim's algorithm. Prim's algorithm initializes the tree M by arbitrarily selecting one node from N; we can assume this initial node is the root R of the given tree T. Then the algorithm repeatedly chooses a node v from M and a node u from N but not in M such that $D(u, v)$ is minimal and connects u to v in M. This continues until all nodes have been connected to M.

We will prove by induction on the size of M that at every stage M is a subtree of T. Clearly, this is true at initialization, when M contains only the root node R. Now, let M be a subtree of T with root R such that $\|M\| \geq 1$, and let u be the next node chosen from N by Prim's algorithm.

If we assume u is not a child, according to T, of any node in M then u must nevertheless be a descendent of at least one node in M (the root R, if nothing else). Let v be the nearest ancestor of u in M. Then there also exists a node q not in M such that q is a child of v and the path from v to u passes through q. Then by assumption

$$D(v, q) < D(v, u)$$

But then, Prim's algorithm would not have chosen u before q for addition to M. Therefore, u must be a (direct) child of some v in T. Also, the path from u to any node $z \neq v$ in M passes through v and by assumption

$$D(u, v) < D(u, z) \quad \forall\, z \in M \; and \; z \neq v$$

So Prim's algorithm must connect u to v in M. Therefore, at every step, Prim's algorithm connects some node not in M to its parent according to T, which is already in M. Thus, at every step, M is a subtree of T, and at the end of Prim's algorithm, the two must be the same.

4.2 Finding a Weighted Euclidean Metric via Mean Square Error Minimization

Theorem 4.1 gives us a set of conditions on a pairwise distance metric so that the minimum spanning tree matches a given tree. Here, we define a weighted Euclidean distance metric and then propose a set of linear equations on the weights by assigning a constant value to the

distance between each two nodes. We then propose a mean square error minimization approach to search for a solution for the weights that satisfies these equations. Let n be the number of cells in the differentiation tree we aim to reconstruct and let m be the length of the gene expression profile of each cell in the tree. For simplicity, we represent the cells by the set of nodes $N = \{1, 2, \dots\dots, n\}$ where each node i is associated with an m-dimensional expression vector $x_i = (x_{i,1}, x_{i,2}, \dots\dots, x_{i,m})$.

Let $\mathbf{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix}$ be a vector of non-negative weights assigned to the genes.

We define a weighted Euclidean distance metric $D: X \times X \to R_0^+$ over a vector space X of dimension m such that for any two vectors $x_i, x_j \in X$

$$D(x_i, x_j) = \sqrt{\sum_{c=1}^{m} (x_{i,c} - x_{j,c})^2 w_c} \qquad (4.2)$$

If we satisfy the condition in (4.1) for every three nodes i, j and k in N where j is on the path from i to k in T, then the minimum spanning tree of the nodes N is precisely the tree T. There are many sets of pairwise distances that we can choose to satisfy (4.1) for all nodes. Perhaps most obviously, the tree distances themselves--i.e. the number of links in the shortest path in the tree T between every two nodes. But of course, many other choices are possible too. If we somehow choose a pairwise distance matrix *D*, how can we construct a weighted Euclidean metric, as a function of expression, consistent with *D*?

If we choose $\mathbf{T} = [t^2(1,2)\ t^2(1,3) \dots\ t^2(1,n)\ t^2(2,3)\ t^2(2,4) \dots t^2(2,n) \dots\dots t^2(n-1,n)]$ to be the vector of pairwise distances $t(i,j)$ that satisfy the condition in (4.1) for all nodes in T such that $t(i,j) = t(j,i)$ then the minimum spanning tree of the nodes N will be precisely the tree T if we are able to find a set of weights **w** such that

$$D(x_i, x_j) = t(i,j) \qquad for\ all\ i\ and\ j\ in\ N \qquad (4.3)$$

Then using the definition of our weighted distance metric in (4.2), we can write (4.3) as follows after squaring both sides of the equality

$$\sum_{c=1}^{m}(x_{i,c}-x_{j,c})^2 w_c = t^2(i,j) \qquad (4.4)$$

If we consider all (i,j) node pairs in N where $i<j$, we end up with a set of linear equations in the form of (4.4) that can be grouped into the following matrix form

$$\mathbf{Cw} = \mathbf{T}^T$$

Where $\mathbf{C} \in \mathbf{R}^{p\times m}$ and $p = \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2}$

If there is any vector of weights **w** satisfying these equations, they will provide a weighted Euclidean metric that will, by Theorem 4.1, make the given tree *T* a minimum spanning tree. However, there may be more than one vector **w** that satisfies these equations. One of those weight vectors, if any exist, can be found by solving the following mean square error minimization problem:

$$\min_{\mathbf{w}} ||\mathbf{Cw}-\mathbf{T}^T||_2^2$$

4.3 Finding a Weighted Euclidean Metric via Linear Programming

Our choice of the vector **T** in the MSE minimization approach is arbitrary**.** Therefore, finding a weight vector **w** that satisfies all equations for our choice of **T** is not guaranteed. On the other hand, the arbitrariness is dissatisfying with the lack of any justification for our choice. Here, we propose a new group of constraints that do not predetermine the distance between each pair of objects as long as they satisfy the condition in (4.1), effectively allowing for any choice of the vector **T** that satisfies (4.1) for all nodes. Moreover, we propose a linear programming approach seeking a sparse solution for the weights while satisfying these constraints.

Using the definition in (4.2) for our weighted distance metric, (4.1) can be written as follows after squaring both sides of the inequality

$$\sum_{c=1}^{m}(x_{i,c}-x_{j,c})^2 w_c < \sum_{c=1}^{m}(x_{i,c}-x_{k,c})^2 w_c \qquad (4.5)$$

Moving all terms to the left side, we rewrite (4.5) in the following form

$$\sum_{c=1}^{m}\left[(x_{i,c}-x_{j,c})^2-(x_{i,c}-x_{k,c})^2\right]w_c < 0 \qquad (4.6)$$

To make sure that the condition in (4.1) is true for any nodes i, j and k where node j is on the path from node i to node k, it is sufficient that it is true for the cases when k is an immediate neighbor of j since, by transitive law, satisfying (4.1) for i, k and its immediate neighbours as well will take care of the other cases where k is not an immediate neighbor of j. Therefore we end up with a set of linear constraints in the form of (4.6) for each node i in T. These constraints can be grouped into the following matrix form

$$\mathbf{A}_{\mathrm{i}}\mathbf{w} < 0$$

Again, if there is any vector of weights **w** satisfying these constraints, they will provide a weighted Euclidean metric that will, by Theorem 4.1, make the given tree T a minimum spanning tree. Those weights, however, may not uniquely solve the set of constraints. Following a common heuristic in distance metric learning, we propose to favour sparse solutions−those that put positive weights on fewer genes. This can be done by minimizing the sum of weights for all genes, while making sure the constraints of each node in T are satisfied by solving the following linear program:

$$\begin{array}{ll} \text{Minimize} & \mathbf{1}^T\mathbf{w} \\ \text{Subject to} & \mathbf{A}\mathbf{w} \le \mathbf{b} \\ & \mathbf{w} \ge \mathbf{0} \end{array}$$

Where $\mathbf{A} = \begin{bmatrix}\mathbf{A}_1 \\ \vdots \\ \mathbf{A}_{\mathrm{n}}\end{bmatrix}$ and where $\mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$ can be chosen arbitrarily, as long as $b_1, \dots, b_m < 0$.

As we show later on, the number of rows/constraints in **A** scales as the square of the number of nodes n and is exactly equal to $(n-1)(n-2)$. If we solve this program as a binary integer program (where weights must be zero or one), it will identify a minimum-size set of genes whose (un-weighted) Euclidean distance makes T a minimum spanning tree. In other

words, it produces a maximally sparse solution. However, binary integer programming is in general an NP-hard class of problem (Papadimitriou 1981 and Garey et al. 2002). If we solve this as an ordinary linear program (where weights are real-valued), solutions are known to favour a limited number of nonzero weights, although the number is not an absolute minimum. The advantages of the ordinary linear program are computational tractability and the fact that genes can be evaluated based on the relative weights they receive, which is not possible with the integer program where all selected genes receive a full weight of one. The choice of binary versus ordinary linear programming solution thus depends on one's preference for the type of solution and computational feasibility.

To better understand the practical computational complexity of the linear program, it is important to understand the size of the constraints matrix. We note that the individual constraints are not sparse. Every weight appears in every constraint, except when by chance the multiplying factor in (4.6) is zero. Thus, the key question is how many constraints there are. The constraints require that for each pair of nodes i and j in the tree, the distance between i and j is smaller than the distance between i and each neighbor of j except for the neighbor that is on the path between i and j. Therefore the total number of constraints $C(T)$ for a given tree T with n nodes can be calculated with the following formula:

$$C(T) = \sum_{i=1}^{n} \sum_{j=1}^{n} Neighbors(j) - 1 \qquad i \neq j \qquad (4.7)$$

In a hierarchical tree the neighbours of a node are its children and its parent, with the exception of the root whose number of neighbours is equal to the number of its children given that it has no parent. If we reorder the nodes so that node 1 represents the root of the differentiation tree then (4.7) can be written a follows:

$$\begin{aligned} C(T) &= \sum_{i=2}^{n} \{Children(1) - 1\} + \sum_{i=1}^{n} \sum_{j=2}^{n} Children(j) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} Children(j) - \sum_{i=2}^{n} 1 \qquad i \neq j \end{aligned} \qquad (4.8)$$

In the first summation in (4.8), the children of each node j in the tree are counted $n-1$ times. Since each node in the tree, except for the root, is a child of one and only one node

then the first summation resembles counting each node in the tree other than the root $n-1$ times. Therefore the result of the first summation is $(n-1)(n-1)$ and the result of the second summation is $(n-1)$. Hence, (4.8) can be written as follows:

$$C(T) = (n-1)(n-2) \tag{4.9}$$

The main advantage of the linear programming approach over the MSE minimization approach is that it does not aim at a specific target distance between each two nodes but rather at any distance that does not violate the conditions set by Theorem 4.1. Therefore, the search domain is larger and the chances of finding a weight vector, if any exist, are higher. The next step was to test these two approaches on the gene expression data for the differentiation tree in Figure 1.

5 IMPLEMENTATION, TESTING AND EVALUATION

In this chapter we apply the two approaches for finding a weighted Euclidean distance metric described in Chapter 4 on the gene expression data of the tree in Figure 1. We first show that solving the MSE minimization problem with the objective of setting the distance between each two nodes in the tree to be equal to the length of the path (i.e. number of edges) connecting them does not produce a valid metric. We then show that the linear programming approach does produce a valid weighted Euclidean metric on just 175 genes hence achieving our goal of reconstructing the correct tree using a very small number of genes. However, we find that there are many other sets of genes that are capable of reconstructing the differentiation tree. Therefore, in the style of random forest training, we construct other random metrics on random subsets of the genes and we evaluate each gene based on how often it receives a nonzero weight in those metrics. By examining the scores of all genes we can see which important genes were omitted by the 175-gene metric and how important are the genes that were included in that metric from an empirical perspective. Finally, since time did not allow us to expand our LP method and validate it on other real gene expression data sets, we test our method on a simpler but related problem: We construct many random matrices and randomly select a few columns from each matrix to construct minimum spanning trees. We then use our LP method of to try and identify those selected columns. The results show that our method performs well, sometimes extremely well, on matrices of specific sizes although it was not designed for that specific problem.

5.1 Solving the MSE Minimization for the Hematopoietic Differentiation Tree

To search for a solution to the MSE minimization we first needed to assign a value to the target distance vector **T.** One way of doing this is to set the target distance $t(i,j)$ between every two nodes i and j in N to be equal to the number of edges on the path connecting them in T. There is no guarantee that the minimization problem will converge to our choice of **T** although other choices may lead to a solution if chosen correctly. We first tried to solve the MSE problem with $(38 \times 37)/2 = 703$ constraints using the MATLAB solver by restricting weights to be positive. The solver was able to find a metric where 69 genes received a positive weight and the remaining 22146 genes received zero weight. However, these

weights resulted in a huge residual norm of 2.3445 x 10^4 indicating that the minimization did not converge to our choice of **T**. Although our constraints are sufficient for tree reconstruction, they may not all be necessary. Hence, a metric that yields a nonzero residual norm may still produce the correct tree. Unfortunately, the residual norm in this solution was too large and therefore the resulting metric did not produce the correct tree (Figure 24).

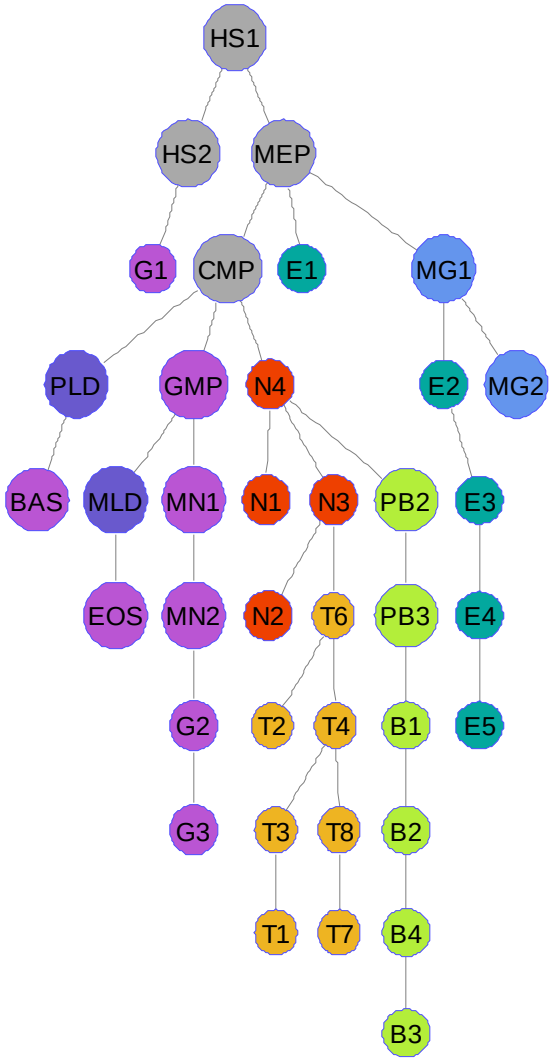


Figure 24 Minimum Spanning Tree Produced by MSE Minimization
The vector of 69 positive weights found by the MSE minimization resulted in a residual norm of 2.3445 x 10^4. The constructed tree does not match with the correct differentiation tree.

We then tried to solve the MSE problem again but this time allowing negative weights in the solution. We obtained a residual norm of nearly zero thus producing the correct tree with

368 positive weights and 335 negative weights and the remaining 21512 genes receiving zero weight. However, with negative weights, the weighted "metric" does not satisfy three of the four axioms of a valid distance metric anymore. With negative weights, the summation under the square root in (4.2) may give a negative result leading to an imaginary distance value which violates the non-negativity axiom of a valid metric. The distinguishability axiom is also violated because, with negative weights, the distance between two distinct objects may be zero. We can also show with a simple example that the triangle inequality axiom can also be violated with negative weights. Consider the three objects a, b and c with the coordinates (1, 3), (2, 4) and (7, 9) respectively and the weighted Euclidean metric D with the weights 1 and -1 on the first and second coordinate respectively. Then, the pairwise distances are $D(a,b) = 0$, $D(b,c) = 0$ and $D(a,c) = 5.29$ so that $D(a,b) + D(b,c) < D(a,c)$, which violates the triangle inequality axiom. Therefore, the weight vector with negative weights provided by the MSE minimization is not a valid solution to our problem.

5.2 A Weighted Euclidean Metric on 175 Genes can Reconstruct the Tree

After failing with the mean square error minimization approach, we tried finding a weighted Euclidean metric that could reproduce the correct tree by solving the linear program described in Section 4.3, using the program lp_solve (Berkelaar et al. 2007). Our differentiation tree has $n = 38$ nodes and each node is associated with a gene expression profile of length 22215. Therefore, the matrix $\mathbf{A}$ in our LP has $31 \times 30 = 1332$ rows and 22215 columns. For the inequality constraints, we chose $\mathbf{b} = -\mathbf{1}^T$. Any other negative constant $\alpha\mathbf{b}$ will produce the same weight distribution produced by $\mathbf{b} = -\mathbf{1}^T$ but with all weights scaled by α since, if $\mathbf{Aw} \leq \mathbf{b}$ is true for some vector $\mathbf{w}$ then $\mathbf{A}(\alpha\mathbf{w}) \leq \alpha\mathbf{b}$ is also true. Also, if $\mathbf{1}^T\mathbf{w}_1 < \mathbf{1}^T\mathbf{w}_2$ is true for two feasible vectors $\mathbf{w}_1$ and $\mathbf{w}_2$ then $\mathbf{1}^T(\alpha\mathbf{w}_1) < \mathbf{1}^T(\alpha\mathbf{w}_2)$ is also true. Therefore, if a weight vector $\mathbf{w}$ satisfies the constraints $\mathbf{Aw} \leq \mathbf{b}$ with minimal weight then the scaled vector $\mathbf{w}_s = \alpha\mathbf{w}$ satisfies the constraints $\mathbf{Aw}_s \leq \alpha\mathbf{b}$ with minimal weight as well. However, attaching different constants to different constraints might produce different results. With $\mathbf{b} = -\mathbf{1}^T$ we found a solution with 175 genes receiving a non-zero weight and the remaining 22040 genes receiving zero weight. A list of the 175 genes with non-zero weights is available in the appendix and in the spreadsheet

"Ghadie_Mohamed_2015_excel1.xlsx" as part of the supplementary material for this thesis. Information provided in columns F-R of the spreadsheet was taken from the file "GPL4685-29295.xlsx" that is available with the data set of Novershtern et al. (2011) at http://www.ncbi.nlm.nih.gov/geo/ with accession number GSE24759.

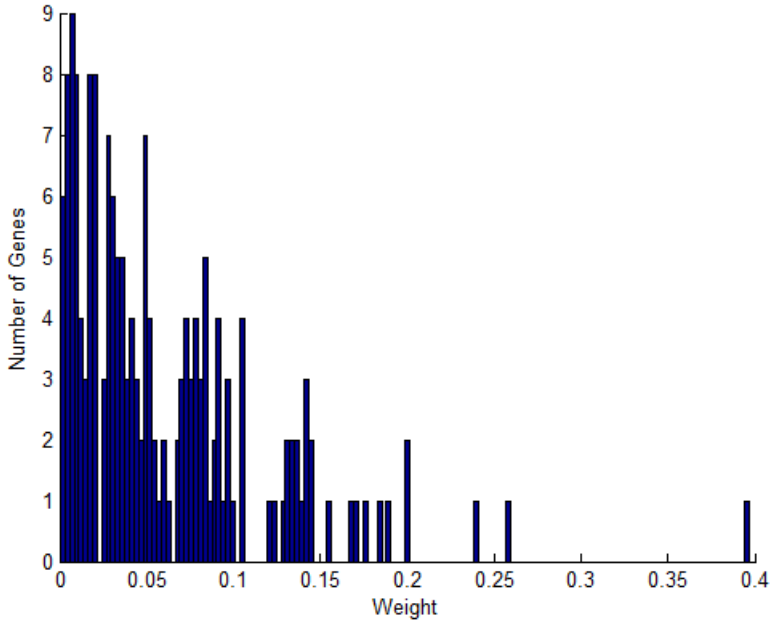


Figure 25 Distribution of the 175 Positive Weights Produced by the Linear Program
The weight vector resulted in successful reconstruction of the differentiation tree.

By looking at the distribution of the 175 nonzero weights in Figure 25 we see that the majority of the weights were less than 50% of the maximum weight. As a matter of fact, out of the 175 weights, 170 fell below 50% of the maximum weight, which means that the genes associated with these weights are not equally important to the distance metric. The 175 genes bore little relationship to the 29 marker genes highlighted in Figure 1. Indeed, only three of those genes appeared among the 175: CD8b1 with 43rd largest weight, CD8a with 60th largest weight, and CD123 with 164th largest weight. Instead, these 175 genes represent an alternative, novel characterization of the different branches of the hematopoietic differentiation tree. Figure 26 shows a heatmap of the expression of the 66 largest-weight genes. Figure 27 shows the heatmap of the expression of all 175 genes. Why we show expression of the 66 largest-weight genes in a separate figure is explained later. Most

weighted genes are expressed primarily in one or a few lineages. For example, the largest weight goes to the Erythropoietin receptor gene (EpoR), which is primarily expressed along the erythroid lineage, where it plays an important signalling role. The next largest weight goes to ZNF198, which is notably absent in erythroid cells, but present primarily in T- and B-cell lineages, and the third largest weight goes to CD1C , which is primarily present in the monocytes and granulocytes and less present in the early erythrocytes and notably absent in all other cell types. If we look at expression of all 175 genes in Figure 27 we see that every gene is primarily expressed in parts of the tree and absent in other parts. Although in both Figures 26 and 27, hierarchical clustering was applied to the un-weighted expression data of the genes, the dendrograms in both figures correctly group most cell types into the right clusters. For example, in Figure 26 when only the 66 largest-weight genes were used, the B-cells formed their own cluster, and the two T-cell groups each formed their own cluster. The late erythrocytes E2-4 also formed one cluster while the early erythrocyte E1 associated itself with the progenitors MEP and CMP and the early stem cells HSC1 and HSC2. This association between stem cells and early erythrocytes was also evident in the results of Novershtern et al (2011). We expect, though, to obtain better clustering results if the expression data of each gene is weighted by the weight it received in the 175-gene solution to the LP. However, we choose not to apply the weights to the clustered data because our goal is to emphasize the actual expression pattern of each gene in different parts of the tree.

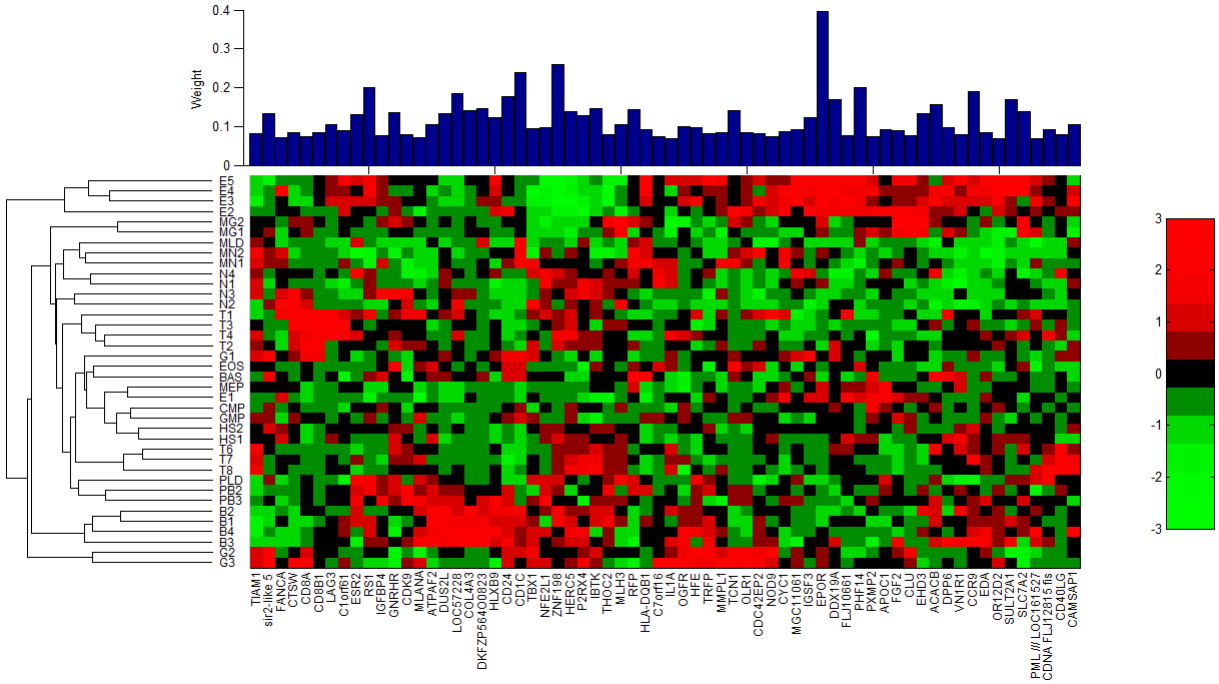


Figure 26 Expression of the 66 Largest-weight Genes in the 175-gene Solution
Heatmap was generated by hierarchical clustering with average linkage on the rows and columns using unweighted Euclidean distance.

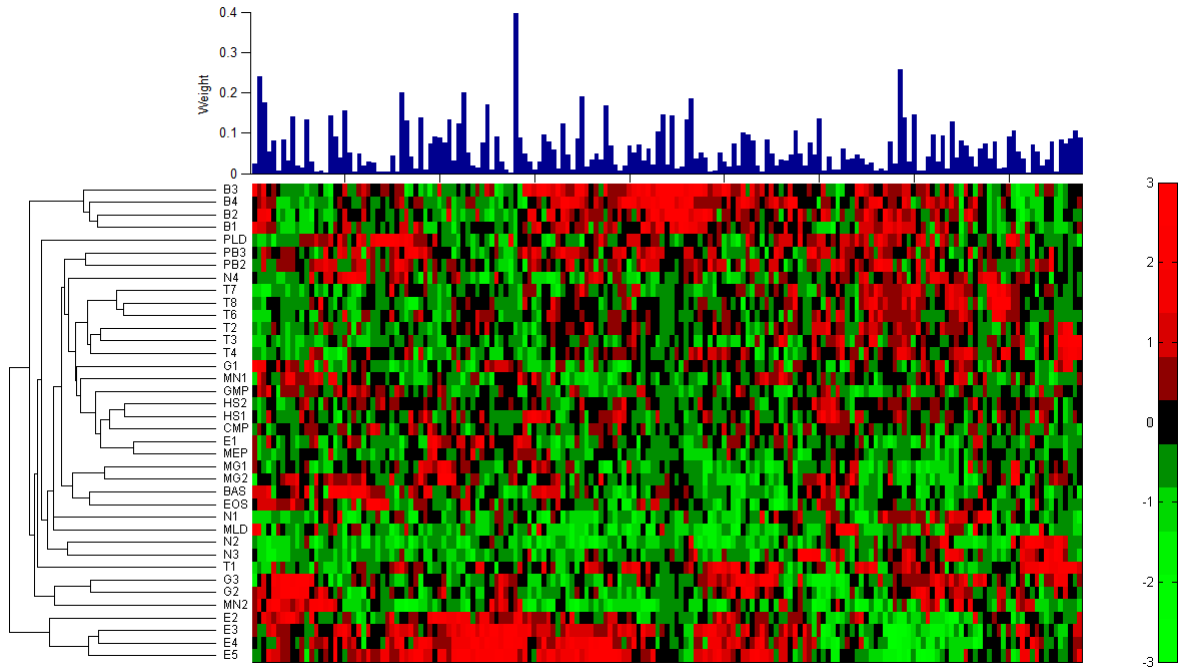


Figure 27 Expression of all 175 Genes in the 175-gene Solution
Heatmap was generated by hierarchical clustering with average linkage on the rows and columns using unweighted Euclidean distance.

To investigate more whether the 175 genes selected by the linear program have any biological significance, we performed a functional annotation analysis using the Database for Annotation, Visualization and Integrated Discovery (DAVID) software (Huang et al. 2009). The functional annotation tool provided by the DAVID software uses the “DAVID Gene Concept”—a graph theory evidence-based method that uses multiple public genomic resources including NCBI (National Center for Biotechnology Information), PIR (Protein Information Resource) and others to agglomerate tens of millions of species-specific gene/protein identifiers from over 65,000 species. Given a list of genes, each associated with a set of annotation terms, the functional annotation tool groups genes that share similar terms by Kappa statistics and provides a list of Gene Ontology (Ashburner et al. 2000) and other biological terms that are most related to the genes in the submitted list. The tool also ranks functional categories based on their co-occurrence with sets of genes in the submitted gene list to unravel biological processes associated with cellular functions and pathways. After submitting our list of 175 genes, the annotation tool returned a long chart of 269 terms but we highlight those that showed the greatest enrichment in Figure 28. The full results of the analysis are provided in the spreadsheet “Ghadie_Mohamed_2015_excel2.xlsx” as part of the supplementary material for this thesis.

The Gene Ontology biological process terms that showed the greatest enrichment were "defence response", "inflammatory response", "response to wounding", and "immune response"---all very relevant and plausible terms, given the presence of the immune-lineage cells in the blood. Next on the list were "regulation of cell proliferation" and "negative regulation of cell proliferation"; these have obvious relevance to both intermediate cell types, where proliferation is typically ongoing, and differentiated cell types, which are typically in a quiescent state. Many subsequent terms involve protein phosphorylation, no doubt reflecting the importance of cell signalling pathways in mediating differentiation decisions. The most significant KEGG (Kanehisa & Goto 2000) pathway was "Hematopoietic cell lineage", followed by several cell signalling and immune-related pathways. CD-family (cluster of differentiation, Zola et al. 2007) genes are used by scientists as markers to discriminate different parts of the hematopoietic tree. Although none of the most-standard

biomarkers appeared on the list of 175, a number of other CD-family or related genes did, including CD1C, CD8A, CD8B1, CD24, CD40LG, etc. Another large category of selected genes (23 of the 175 genes) were transcription factors and/or had GO annotations implicating a role in transcriptional regulation. From multiple points of view, then, the genes selected by the linear program are not just some random set that happens to allow discrimination between the different cell types in the differentiation hierarchy--a potential concern when so many genes are available to choose from. Rather, they clearly reflect major processes and pathways active in different parts of the tree, and to some extent rediscover discriminative families (e.g. the CD genes) upon which scientists also rely.

Sublist	Category	Term	RT	Genes	Count	%	P-Value
☐	GOTERM_BP_FAT	defense response	RT		23	14.8	2.3E-8
☐	GOTERM_BP_FAT	immune response	RT		22	14.2	7.3E-7
☐	GOTERM_BP_FAT	response to wounding	RT		19	12.3	1.1E-6
☐	GOTERM_BP_FAT	inflammatory response	RT		15	9.7	1.2E-6
☐	GOTERM_CC_FAT	intrinsic to plasma membrane	RT		29	18.7	5.7E-6
☐	GOTERM_CC_FAT	integral to plasma membrane	RT		28	18.1	1.1E-5
☐	SP_PIR_KEYWORDS	immune response	RT		10	6.5	7.8E-5
☐	SP_PIR_KEYWORDS	glycoprotein	RT		55	35.5	1.9E-4
☐	GOTERM_BP_FAT	regulation of cell proliferation	RT		19	12.3	2.1E-4
☐	SP_PIR_KEYWORDS	phosphoprotein	RT		80	51.6	3.1E-4
☐	GOTERM_CC_FAT	plasma membrane part	RT		37	23.9	3.8E-4
☐	GOTERM_BP_FAT	negative regulation of cell proliferation	RT		12	7.7	3.9E-4
☐	SP_PIR_KEYWORDS	transmembrane protein	RT		15	9.7	6.2E-4
☐	KEGG_PATHWAY	Hematopoietic cell lineage	RT		7	4.5	7.3E-4
☐	GOTERM_BP_FAT	cell migration	RT		10	6.5	8.3E-4
☐	GOTERM_BP_FAT	positive regulation of protein kinase activity	RT		9	5.8	8.8E-4
☐	GOTERM_BP_FAT	positive regulation of kinase activity	RT		9	5.8	1.1E-3
☐	GOTERM_BP_FAT	positive regulation of response to stimulus	RT		9	5.8	1.3E-3
☐	BBID	58.(CD40L) immnosurveillance	RT		4	2.6	1.4E-3
☐	UP_SEQ_FEATURE	topological domain:Extracellular	RT		37	23.9	1.4E-3
☐	GOTERM_BP_FAT	positive regulation of transferase activity	RT		9	5.8	1.4E-3
☐	UP_SEQ_FEATURE	topological domain:Cytoplasmic	RT		43	27.7	1.7E-3
☐	GOTERM_BP_FAT	cell motility	RT		10	6.5	1.7E-3
☐	GOTERM_BP_FAT	localization of cell	RT		10	6.5	1.7E-3
☐	KEGG_PATHWAY	MAPK signaling pathway	RT		11	7.1	1.9E-3
☐	SP_PIR_KEYWORDS	extracellular protein	RT		4	2.6	1.9E-3

Figure 28 DAVID Functional Annotation Analysis Results for the 175 Genes
Only the highest enrichment terms from the full chart of 269 terms are shown.

5.3 Further Reducing the Number of Genes in the Distance Metric

The linear program selected 175 genes that allow for the reconstruction of the tree, an impressively small number given the large number of genes in our data set. To better understand the significance of this number, we estimated the chances of finding a weight vector that can reproduce the correct tree from a fixed-size set of genes randomly sampled from the data. In more detail, we randomly selected a fixed-size sample of genes from the data 100 times and we tried to solve the linear program on the set of genes selected in each iteration. We then estimated the probability of finding a weight vector for that sample size by counting how many of the 100 iterations generated a feasible linear program. We then repeated this estimation for different gene sample sizes and obtained the values shown as dots in Figure 29. We were then able to fit the points to the logistic function $f(x) = \frac{1}{1+e^{ax+b}}$ scaled by a factor of 100 using $a = -0.0022$ and $b = 14.5279$ which were found by minimizing the mean square error between the fitting function and the measured points. We only start to get feasible programs when the number of genes exceeds 4000 ($\sim$18% of the total number of genes), thus the chance of reconstructing the hematopoiesis differentiation tree of Figure 1 using 175 random genes is practically zero. The probability of getting a feasible program with more than 4000 genes, and therefore successful tree reconstruction, increases at a rate of 5.03×10^{-4} per added gene and reaches 1 at about 9000 genes ($\sim$40% of the number of genes in the data set). Therefore, if we are to randomly choose the genes from the data set, we need a fairly large number (9000) to guarantee successful tree reconstruction only after we solve the LP on those genes. Successful tree reconstruction is still not guaranteed if no weights are applied to the genes.

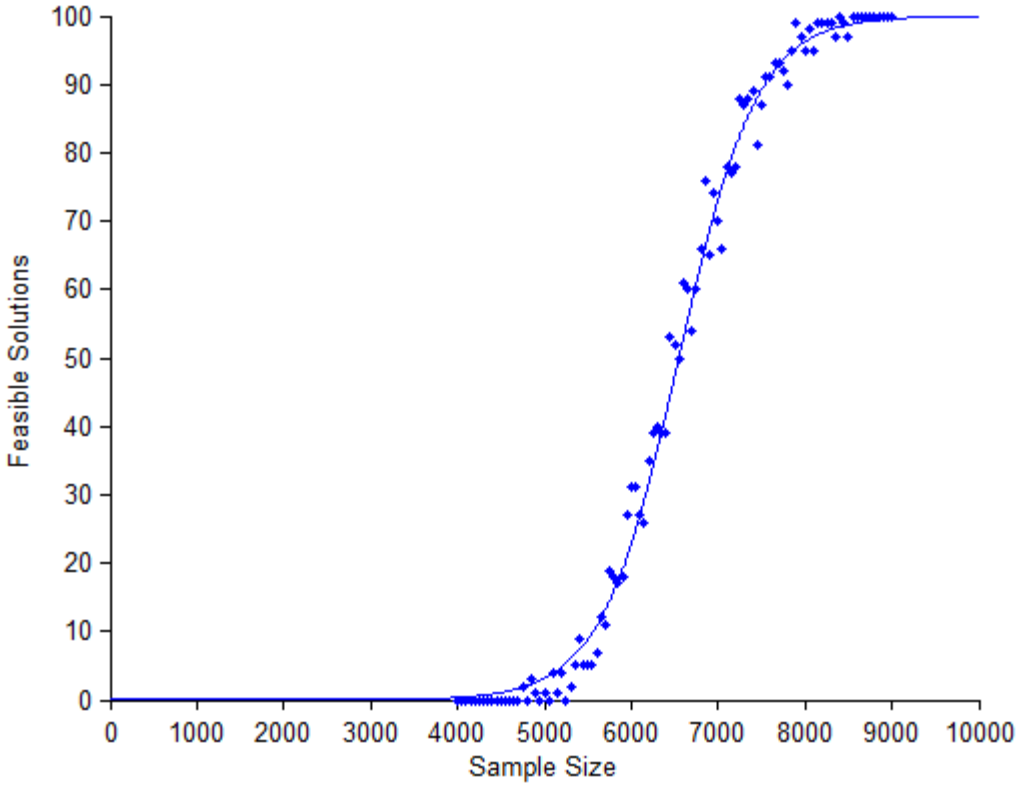


Figure 29 Feasibility of the Linear Program with Gene Subsets of Different Sizes
Feasibility (represented by a dot) for each sample size was calculated by searching for a weight vector 100 times each time with a fixed-size random subset of genes. Measured values were fit to the logistic curve $\boldsymbol{f(x) = 1/(1 + e^{ax+b})}$ scaled by a factor of 100 using $\boldsymbol{a = -0.0022}$ and $\boldsymbol{b = 14.5279}$.

Although 175 is a very small number compared to the total number of genes in the data set, the LP minimizes the sum of weights but not necessarily the number of genes with positive weights. Therefore, we were interested in discovering whether the tree could be reconstructed with an even smaller number of genes. We initially tried to solve the linear program as a binary integer program, hoping to identify a minimum-size set of genes whose Euclidean metric would produce the right differentiation tree. However, due to the large number of variables (22215) and/or constraints (1332) no solver we tried would terminate. Thus, we could neither find a minimum-size solution nor prove infeasibility of the problem. We also tried solving the binary integer version, but restricting attention to just the 27 established marker genes shown in Figure 1. This problem turned out to be infeasible. We

conjecture that nonlinear expression rules (such as the present/absent rules typically used) are necessary to discriminate the branches of the tree based on these genes.

As an alternative, we tried to further winnow down the 175-gene solution. We solved the linear program with inequality constraints $\mathbf{Aw} \leq -\mathbf{1}^T$, but theoretically we only need $\mathbf{Aw} < \mathbf{0}$ to ensure reconstruction of the tree. For n = 1, 2, 3 ... we tried retaining the n largest weights, setting the smaller ones to zero. We found that we could set all weights smaller than the 122nd-largest to zero and still satisfy the latter set of strict inequality constraints. As we show in Figure 30, zeroing out more weights than that resulted in violation of at least one constraint. However, one must keep in mind that our constraints are sufficient conditions, not necessary ones. We further found that with as few as the n = 89 largest weights, although constraints were violated, the induced Euclidean distance metric still resulted in the reconstruction of the correct tree. Finally, we reduced the number of genes even further by restricting attention to genes with the top n weights, but re-solving the linear program using just those genes. We obtained a feasible solution with as few as the top n = 66 genes--the same ones shown in Figure 26. However, the sum of weights in the solution increased gradually as we eliminated genes one by one from the 175-gene group and re-solved the linear program with the same inequality constraints $\mathbf{Aw} \leq -\mathbf{1}^{\mathrm{T}}$ using just the genes left in the group (Figure 31). The relative gap between the largest weight and the smaller weights also shrank as we re-solved the program on fewer genes (Figure 32). It remains to be seen whether any smaller solution can be found.

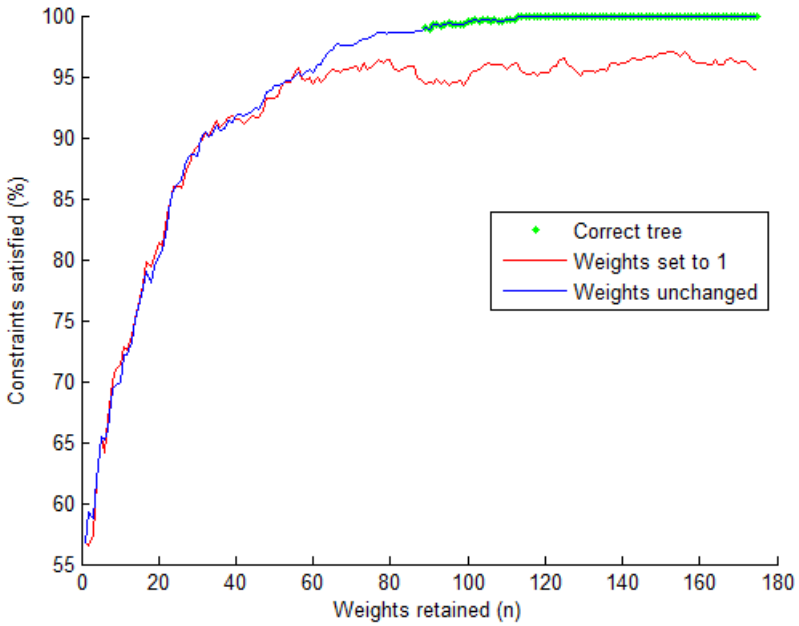


Figure 30 Constraints Satisfied using the *n* Largest Weights in the 175-gene Solution
The *n* retained weights were either set to 1 (Red) or kept unchanged (blue). The differentiation tree could still be reproduced when only few constraints were violated (Green).

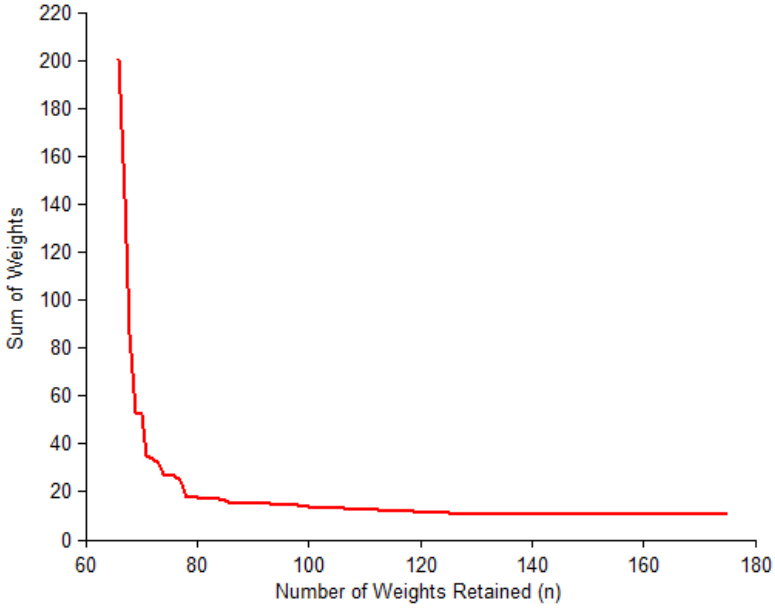


Figure 31 Sum of Weights
Sum of weights calculated after re-solving the linear program on the *n* largest-weight genes in the 175-gene solution

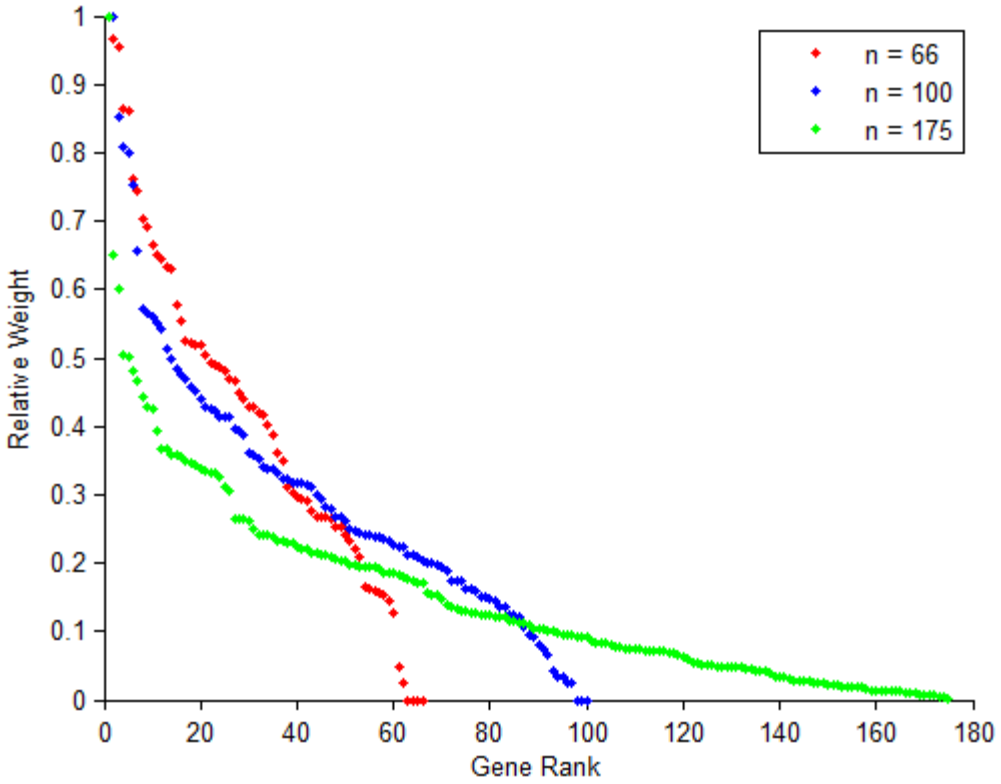


Figure 32 Relative Weights in Relation to Number of Genes Retained
Relative weights after re-solving the linear program on the *n* largest-weight genes in the 175-gene group. Weights in each solution were normalized by the maximum weight.

5.4 Most of the 175 Genes Receive Large Weights in Random Metrics

Minimizing the sum of weights is a heuristic that favours sparse solutions. However, there may be multiple solutions with the same sum of weights, and there may be other solutions with larger weight sums that are still "good" solutions in some sense (such as having a smaller number of nonzero weights). Therefore, to better evaluate the importance of each gene, we proposed an approach similar to that used in Random Forests (Breiman 2001). We solved the linear program for finding a weighted Euclidean metric 100 times, each time on a random fixed-size subset of genes. Intuitively, the more often a gene is given a non-negative weight, and the larger those weights are, the more "important" that gene is to the construction of the distance metric. We chose a sample size of 7000 that is small enough to allow for diversity among the samples and large enough to generate a feasible linear program. Out of the 100 attempts, 70 were successful, with the remainder generating

infeasible linear programs. We then calculated the score S_i for each gene i by integrating its ranks from all 70 feasible solutions using the following equation:

$$S_i = \sum_{k=1}^{F} \left(1 - \frac{r_{ik} - 1}{W_k}\right) \qquad (5.1)$$

Where F is the number of feasible programs, W_k is the number of nonzero weights in the k^{th} solution, and r_{ik} is the rank of gene i among the genes with nonzero weights in the k^{th} solution. The rank of a gene in a particular solution was included in its score calculation only if it received a nonzero weight in that solution.

Out of the 22215 genes, 1573 received a nonzero weight in at least one solution and therefore received a nonzero score although the majority of those genes had low scores (Figure 33). A list of these 1573 genes is available in the spreadsheet "Ghadie_Mohamed_2015_excel3.xlsx" as part of the supplementary material for this thesis. Information provided in columns F-R of the spreadsheet was taken from the file "GPL4685-29295.xlsx" that is available with the data set of Novershtern et al. (2011) at http://www.ncbi.nlm.nih.gov/geo/ with accession number GSE24759. For each gene, we calculated the average of all weights it received in all solutions and its frequency of receiving a nonzero weight. We use the notion of frequency since the 7000 genes that participated in solving each of the 70 feasible linear programs were randomly selected, therefore not all genes were selected an equal number of times. We calculate the frequency for each gene by dividing the number of solutions in which a gene received a nonzero weight by the number of times it was selected among the random 7000 genes. We then plotted the average weight against the frequency of receiving a nonzero weight and we found a positive correlation between the two (Figure 34). This correlation was also evident in gene scores and their corresponding weights in the 175-gene solution (Figure 35).

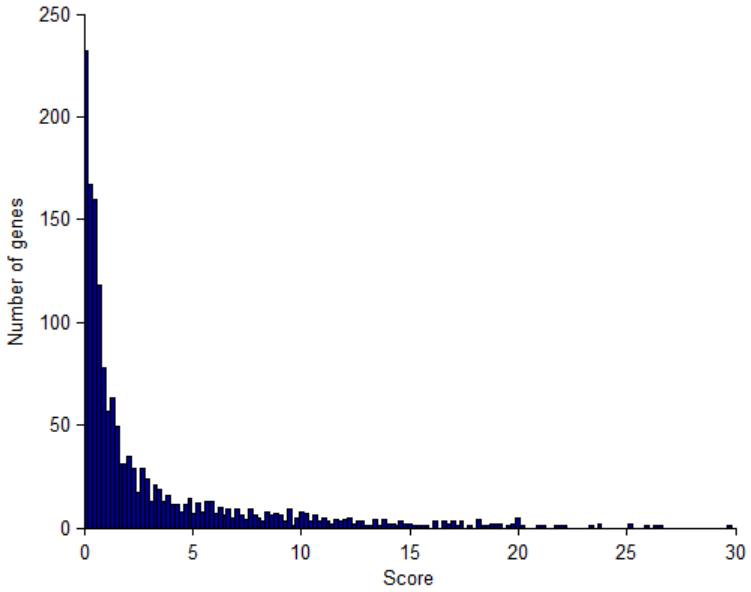


Figure 33 Gene Score Distribution
Distribution of scores for the 1573 genes that received nonzero scores over 70 metrics constructed from size-7000 random subsets of the genes.

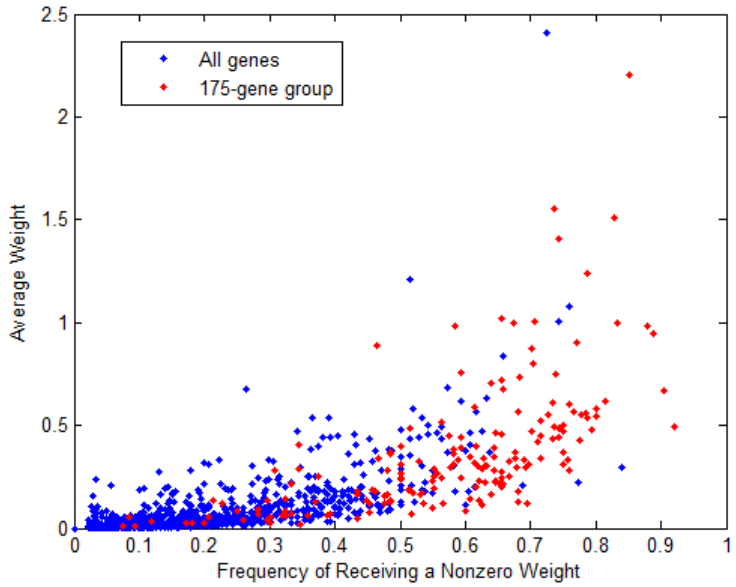


Figure 34 Gene Average Weights in 70 Random Metrics
Average weight in relation to the frequency of receiving a nonzero weight for each gene in 70 metrics constructed from size-7000 random subsets of the genes. Frequency represents the number of solutions in which a gene received a nonzero weight normalized by the number of times it was selected among the random 7000 genes.

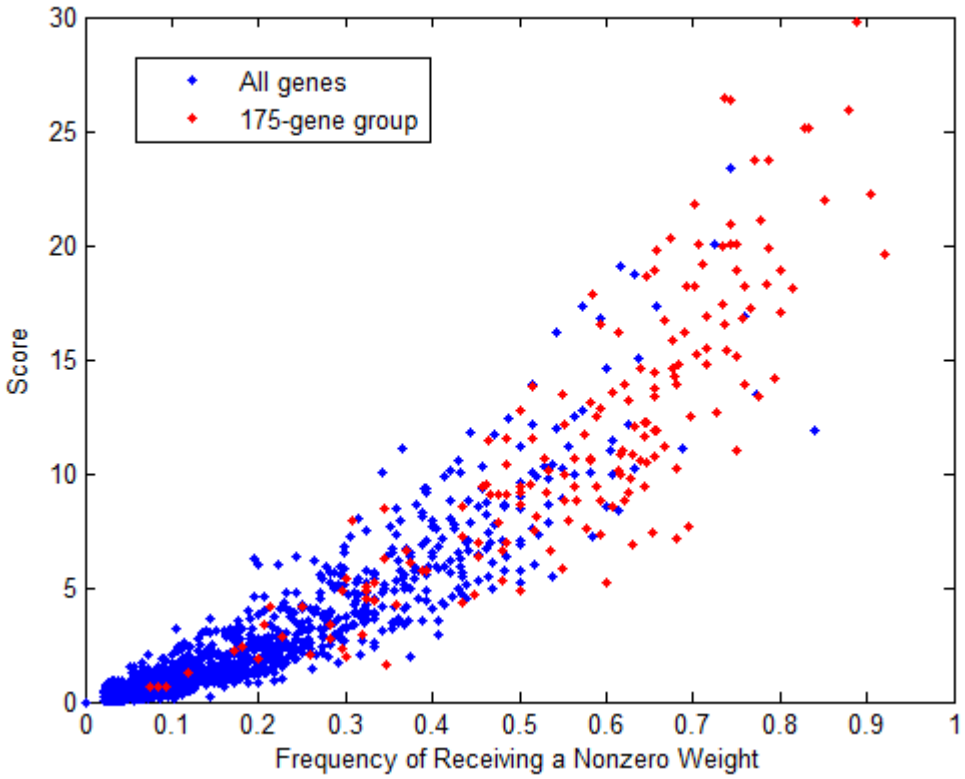


Figure 35 Gene Scores in 70 Random Metrics
Scores in relation to the frequency of receiving a nonzero weight for each gene in metrics constructed from size-7000 random subsets of the genes. Frequency represents the number of solutions in which a gene receives a nonzero weight normalized by the number of times it was selected among the random 7000 genes.

All genes in the 175-gene group received a nonzero score and most of them scored in the top ranks, confirming the significance of these genes from an empirical perspective. However, there are a few genes that received a large average weight and/or a high score but were not selected in the 175-gene group. It may be that a gene receives a nonzero weight when other genes in the subset of 7000 are not too relevant to blood cell functions. However that may not guarantee it a spot in the 175 gene group when competing with other more relevant genes. An example is CNR1, a receptor that is mainly involved in cannabinoid-induced CNS effects and is not significantly relevant to blood cell functions. CNR1 does have the largest average weight however its frequency of receiving a nonzero weight is lower than that of other genes in the 175-gene group and therefore got the 16th highest score. Furthermore, it did not make it into the group of 175 genes where other more relevant genes are present. On

the other hand, the gene with the 2nd largest average weight, COL4A3, which is notably present in B-cells only and whose coded Tumstatin cleavage fragment possesses anti-angiogenic and anti-tumor cell activity, got the 11th highest score and received the 15th largest weight in the 175-gene group. Therefore, some genes may end up with a large average weight if they just happen to receive large weights in a few solutions where other participating genes are not too relevant to the cell types. That, though, does not mean they will receive high scores when their frequency of receiving nonzero weights is taken into account. It also seems that genes with high scores and/or high average weights that were not included in the 175-gene group were left out because there is no need to rely on them in the presence of other more relevant genes in the group. Nevertheless, because our linear program aims to reconstruct the tree with a small number of genes, it would be no surprise if some potentially-relevant genes are omitted. Which relevant genes are excluded from the solution and how to improve on that area is worth more in-depth investigation in the future.

5.5 Identifying Features Used to Construct Trees from Variable-Size Matrices

We were still interested in further evaluating our method empirically but this time on different data and with different measuring criteria. We thus decided to try to solve a different problem that might be useful in other applications as well. We proposed the following question: Given a random $n \times m$ matrix M whose rows correspond to nodes and columns correspond to features of those nodes, and given a minimum spanning tree T constructed on the rows of M using $p = 1, 2, 3, \ldots$ columns of M, identify those p columns by finding a weight vector for the tree T using the LP approach described in Chapter 4. Examining the program's rate of success in identifying those columns on matrices with different sizes will give us insights into the reliability of the program in relation to the size of the data it is applied to. Moreover, this approach will further address the degeneracy of solutions to the linear program in the sense that if our method is capable of always identifying the single column of a matrix used to build a minimum spanning tree then we should be able to identify the gene whose expression is sufficient to define the structure of a differentiation tree without worrying about other possible solutions, if they exist.

We first wanted to measure the program's performance in relation to the number of columns m in the matrix M. So, we vary m, and for each value of m we produce a certain number of minimum spanning trees each from p columns randomly selected from M and build a linear program from each tree. Although each of those trees is a minimum spanning tree of the rows of M, constructed using different subsets of its columns, some of them, or even all of them, may not satisfy all our LP constraints, specifically the constraints that are unnecessary for a tree to be a minimum spanning tree. Thus, not all programs produced from those trees may be feasible and, as the results later show, some trees may never produce feasible programs. Therefore, we set a goal of 20 feasible programs for each value of m but with a limit of 200 on the total number of trees we are allowed to produce for that m. We may, though, require the maximum number of 200 attempts to produce 20, or even fewer, feasible programs for some values of m. On the other hand, we may require less than 200, or even just 20, attempts to produce 20 feasible programs for other values of m. We then solve all feasible programs for each m and we count how many of the resulting metrics assigned the largest weights to the p columns that were used to construct the tree associated with each program. We describe our approach in more detail below:

Repeat for $m \leftarrow 5\ to\ 500\ in\ steps\ of\ 5$
Initialize $passes \leftarrow 0$
Initialize number of tries: $Tr \leftarrow 0$
Initialize number of feasible programs: $FP \leftarrow 0$

Repeat until Tr reaches 200 or FP reaches 20

1. Increment Tr by 1
2. Generate an $n \times m$ random matrix M from a normal distribution
3. Randomly select p columns of M and build a minimum spanning tree T on the rows of M using an un-weighted Euclidean metric on those p columns
4. Construct the linear program from T using all columns of M
5. If program is feasible, do:
   - Increment FP by 1
   - If the p largest weights were assigned to the same p columns of M, increment $passes(m)$ by 1

When only one column was selected to build the tree, the linear program was able to identify that column, by assigning it the largest weight, for all matrices with 10 rows and fewer than 100 columns (Figure 36). However, the program's performance gradually dropped to 70% as we increased the number of columns to 500. Moreover, the linear program always succeeded in identifying that single column for matrices with 15 and 20 rows, and fewer than 500 columns. We conjecture that this perfect performance would have dropped lower if we were to increase the number of columns beyond 500. It is no surprise, though, that it gets harder to find the correct column as we increase the total number of columns in *M* since the correlation between different columns increases as more columns are added to the matrix. This happens even more often with gene expression data where correlations among genes' expression are very strong.

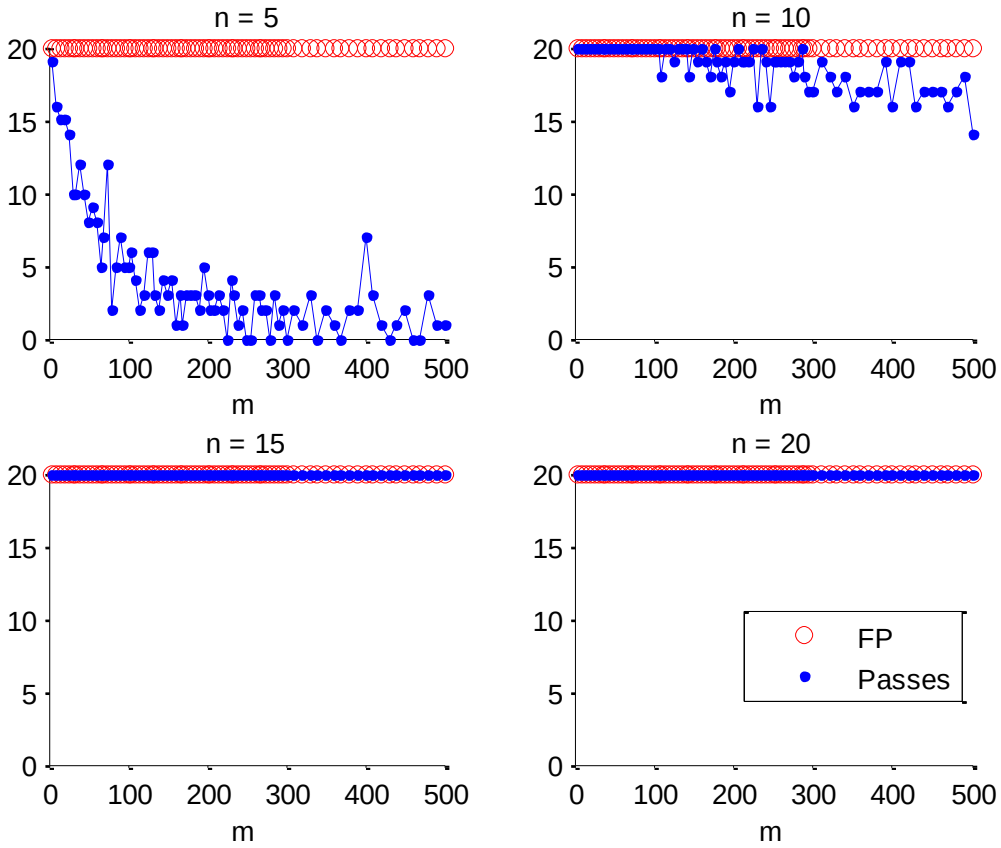


Figure 36 Identifying the Single Feature used to Build a MST
For each *m*, 200 attempts were made to produce 20 minimum spanning trees with feasible linear programs from one column randomly selected from a random *n* x *m* matrix. FP represents the number of trees (maximum of 20) that produced a feasible program. Passes represent the number of feasible programs (FP) that successfully identified the single column used to construct the tree.

We repeated the same experiment but this time increasing the number of selected columns to two and then three. The results in Figure 37 and Figure 38 show that the program's performance was not as good as it was when only one column was selected. We also noticed that the program's performance always reached a peak at some ratio of columns (m) to rows (n), and that peak was lower for higher numbers of selected columns. The peak performance, however, always occurred at a value of m for which we successfully produced 20 feasible programs. In fact, the performance when more than one column was selected is not only lower at the peak but at all values of m where 20 feasible programs were produced. Therefore, we cannot make any correlation, from these results, between program feasibility and the decline in performance as we increase the number of selected columns. The decline in performance, though, may be the result of redundancy/correlation in the data matrix M which has little to do with the LP hence we cannot totally blame it on the LP method. It would be a good option in the future to try to separate the effect of redundancy in the data matrix on these results from the actual performance of the LP method.

Moreover, the ratio of m to n at which the peak performance occurs also increases for matrices with a larger number of rows. For example, when we fixed the number of selected columns to two, the results in Figure 37 show that the peak performance at n = 5 occurred at the lowest value of m with m/n = 1. While that ratio increased to 1.5, 5.33 and 6.5 for n = 10, 15 and 20 respectively, the program maintained that peak performance for a wider range of m as the value of n increased. Therefore, it seems that for a larger number of rows in the data matrix, a larger ratio of columns to rows is needed to optimize the LP's performance. When we increased the number of selected columns to three, the peak ratios did not show much of an increase with m/n = 1, 1.5, 5.67 and 7.75 for n = 5, 10, 15 and 20 respectively, indicating that the m/n ratio for peak performance is not affected much by the number of columns selected to build the tree. However, the peak height is lower and drops faster as m increases when the number of selected columns is increased. It remains to be seen whether these conclusions are true for larger matrices and larger numbers of selected columns.

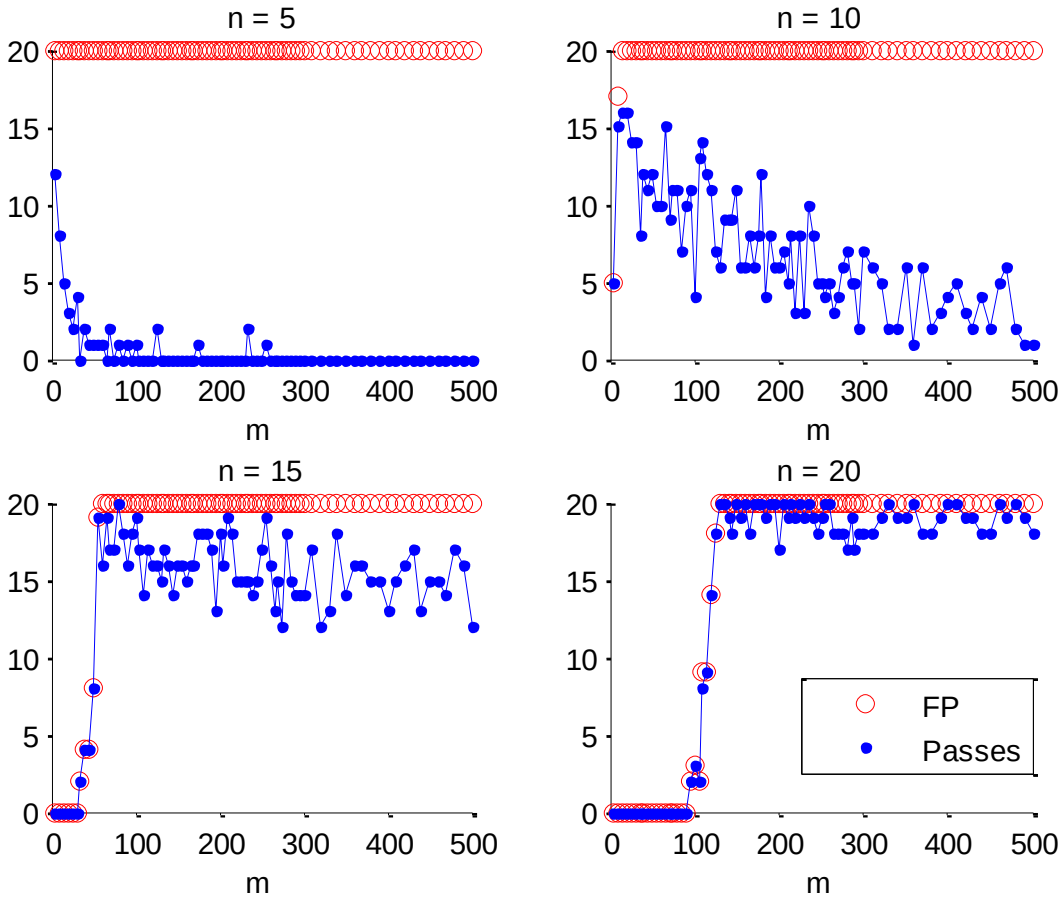


Figure 37 Identifying the Two Features used to Build a MST
For each *m*, 200 attempts were made to produce 20 minimum spanning trees with feasible linear programs from two columns randomly selected from a random *n* x *m* matrix. FP represents the number of trees (maximum of 20) that produced a feasible program. Passes represent the number of feasible programs (FP) that successfully identified the two columns used to construct the tree.

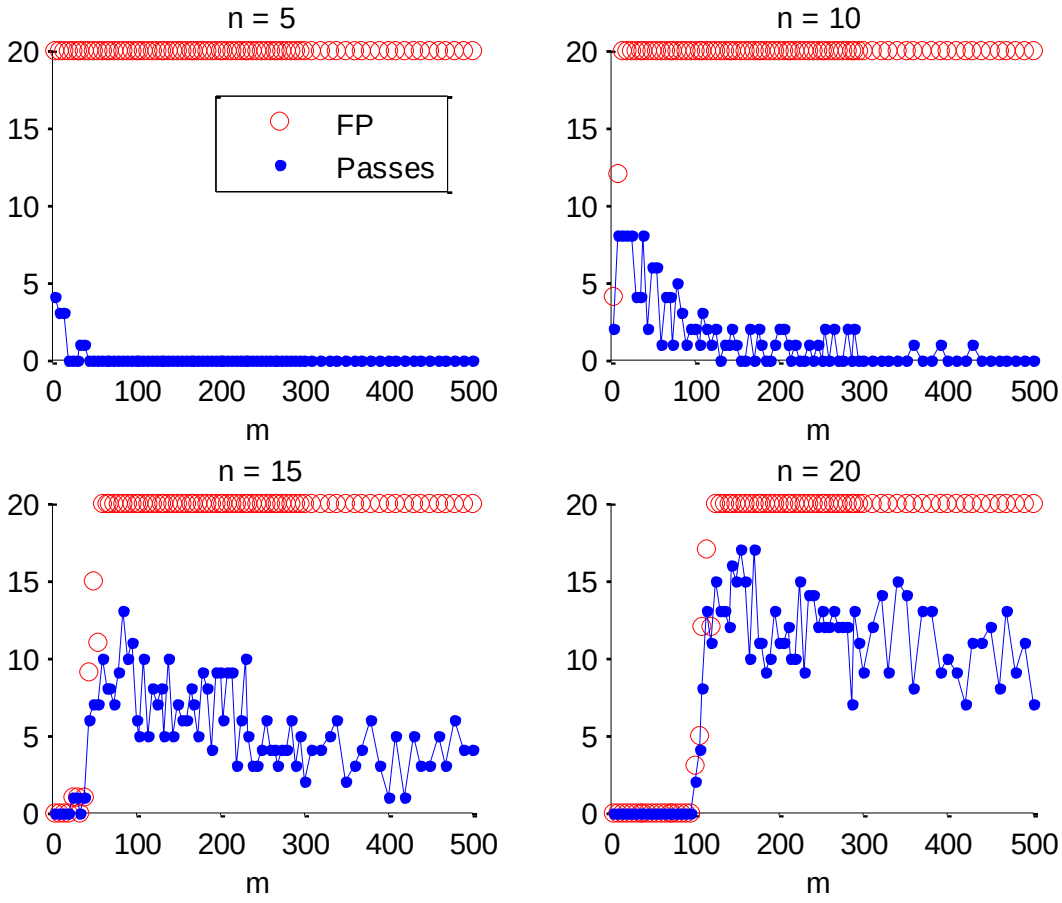


Figure 38 Identifying the Three Features used to Build a MST
For each *m*, 200 attempts were made to produce 20 minimum spanning trees with feasible linear programs from three columns randomly selected from a random *n* x *m* matrix. FP represents the number of trees (maximum of 20) that produced a feasible program. Passes represent the number of feasible programs (FP) that successfully identified the three columns used to construct the tree.

We repeated the experiment but this time on smaller matrices by fixing m to 50 and varying n from 3 to 50. As shown in Figure 39, the program's performance reached its peak at m/n = 5.55 when one column was selected and remained at that level for all values of n. Although the program's performance reached its peak at m/n = 3.85 when more than one column was selected to build the tree, that peak ratio did not change for 2, 3 and 4 selected columns. Again, this suggests that the peak ratio m/n is not affected by the number of columns selected to build the tree, at least when the number of columns selected is larger than 1. However, the results in Figure 39 also show that once the number of rows exceeds 17, the 200 attempts were not able to produce any feasible LPs, which is not a surprise given

that the number of columns is fixed to 50. In this case, we can relate the decline in performance to the infeasibility of the LPs mainly caused by the unnecessary constraints.

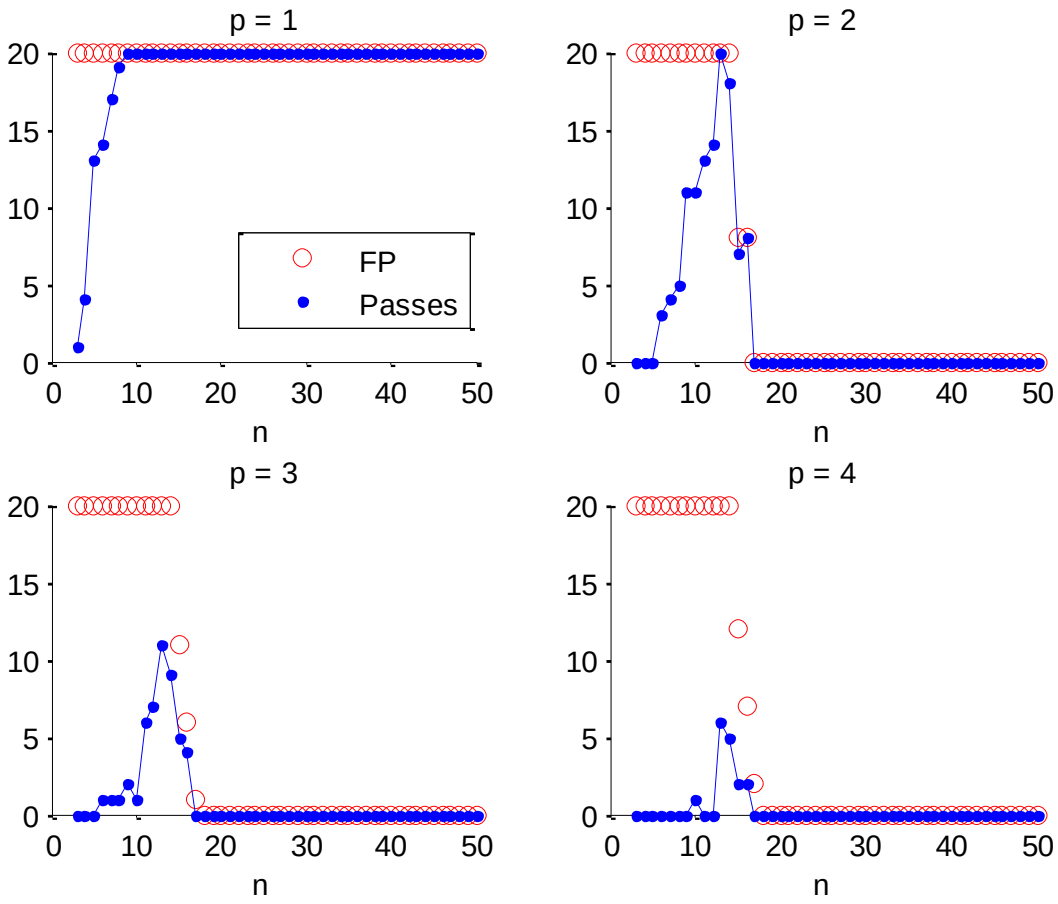


Figure 39 Identifying the Features used to Build a MST on Variable-Size Sets of Nodes
For each *n*, 200 attempts were made to produce 20 minimum spanning trees with feasible linear programs from *p* =1, 2, 3 and 4 columns randomly selected from a random *n* x *m* matrix (*m* = 50). FP represents the number of trees (maximum of 20) that produced a feasible program. Passes represent the number of feasible programs (FP) that successfully identified the *p* columns used to construct the tree.

In this Chapter, we successfully achieved our goal to learn a weighted Euclidean distance metric that allows for reconstruction of the hematopoiesis differentiation tree in Figure 1. Our investigations show that the 175 genes selected by this metric are biologically relevant to the cell types and the structure of their differentiation hierarchy. Further experiments showed that these 175 genes are not just randomly selected genes but are more relevant than other genes from an empirical point of view as well. We also demonstrated how our LP approach can be useful in other related problems such as identifying features that were used

to build MSTs from random data matrices whose column to row ratios fall within a specific range. This latter result illustrates the capability of our method in overcoming the multiple-solution problem when searching for one or few genes that are able to describe the differentiation tree. Last but not least, we have raised several important points that need to be addressed in our LP approach in the future but also open doors to more interesting research.

6 CONCLUSIONS AND FUTURE WORK

Using hematopoietic differentiation as an example we have shown that, besides marker genes which are merely just helpful in identifying different cell types, distinct branches in the differentiation tree can be discriminated using expression of thousands of other genes. These genes offer an alternative way to describe the structure of the tree when their expression pattern along each lineage is classified into four different categories: up-regulated, down-regulated, fluctuating and stable. We have also shown that while traditional methods such as hierarchical clustering and maximization of parsimony fail to reconstruct the correct type of tree for intermediate and fully-differentiated cell types, reconstructing the tree as a minimum spanning tree is possible using a weighted Euclidean distance metric. Our investigations show that genes selected by this metric are empirically significant and relevant to the cell types and biological processes active throughout their proliferation and differentiation. Therefore, our work offers a new way to characterize biologically-meaningful relationships between gene expression and differentiation trees.

The terms on the left hand sides of (4.4) and (4.6) are linear summations of difference terms. Since each term corresponds to a single gene, the value of the total sum in each constraint is insensitive to mean changes resulting from normalizing the expression of each gene over all cell types. However, normalizing the data still results in dividing each difference term in the summation by the variance of the gene it represents; hence dividing each term by a different variance. The 175-gene solution we got from the linear program (like all other results in Chapter 5) is based on the normalized data. When we solved the linear program on the expression data without normalizing, we got 152 genes only with positive weights. Out of those 152 genes, only 88 appeared in the 175-gene group, and those 88 genes were not all in the highest ranks in the 152-gene solution. Therefore, normalizing the gene expression data caused our method to produce a larger solution with nearly half of its genes being different than those in the solution produced from unnormalized data. Whether this effect of data normalization generalizes to any data set and how it relates to the variances of the genes is yet to be investigated.

Our two approaches to finding gene weights that allow for tree reconstruction work by deriving a sufficient set of conditions on the weights of the metric. In the MSE approach, these conditions, if satisfied, ensure that each two nodes/cells in the constructed tree are at a specific distance of our choice. We are aware, though, that even if a weight vector that allows for tree reconstruction exists, the MSE approach will not find that vector or any other feasible vector if we do not choose the correct pairwise distances between the nodes. Even if the method succeeds in finding a weight vector for a given choice of pairwise distances, we cannot guarantee the uniqueness of that solution. However, in the case when no weight vector can produce our choice of distances, the MSE approach is an easy and efficient way to confirm that. We need to keep in mind, though, that since not all conditions are necessary for tree reconstruction, removing one or more of those conditions while keeping the remaining conditions unmodified may ultimately lead to a solution.

Although the inequality constraints we proposed in the LP approach are more relaxed than the equality conditions in the MSE approach, even those constraints are not all necessary to reconstruct the differentiation tree. For the hematopoietic tree specifically, reconstruction was still possible when roughly 99% of the LP constraints were satisfied. It seems unlikely that identifying and eliminating those few unnecessary constraints would substantially change either the resultant weights or the computational burden of finding them. However, it may be that some entirely different constraint formulations could be found, which might generate different solutions and/or make tractable the binary integer program, which would be of great interest. Although those few constraints were found unnecessary for reconstruction of the hematopoietic tree, that does not make them unnecessary for reconstruction of other trees. We also need to keep in mind that those few constraints were violated by eliminating genes one by one from the 175-gene solution until we were left with 89 genes. If we were to use another feasible weight vector as a starting point, it may be that the tree can be reconstructed with a different subset of the constraints. But then, these results would be specific to that tree and its associated expression data. Therefore, it remains to be seen what constraints are necessary for reconstruction of a tree of any predetermined structure regardless of the nature of the data.

One arbitrary choice we made when solving the LP on the Novershtern data was to set the upper limit on each constraint to -1. If we had set different limits on different constraints we may have obtained different weight vectors. It remains to be seen how this degree of freedom in the approach might be utilized. But, it might allow for generation of sparser solutions, or perhaps the incorporation of more detailed prior knowledge about the "distances" between different cell types. Identifying the "best" limit for each constraint and its effect on the results is thus worth investigating as well. Although the weight vector chosen by the LP solver may be unique in the sense that it has a minimum sum of weights, we are aware that other solutions with larger weight sums also exist. This degeneracy property is not to be seen as a design flaw but rather an expected result of gene co-expression. If we were able to eliminate genes with smaller weights from the 175-gene solution then it is most likely that those genes can be replaced by other co-expressed genes or genes with similar biological functions. Our work can therefore be extended to search for solutions limited to genes with common biological functions. For example, what genes are selected, and with what weights, if we restrict attention to known marker genes, or to transcription factors, or signalling pathway genes, or metabolic genes, and so on.

We chose to solve the LP optimization on the tree of Figure 1 using the mean of all replicates for each cell type as the gene expression profile representing that cell type. Although a reasonable choice, one may choose to represent each cell type by only one of its replicates and then solve the LP to find a set of weights that allow for tree reconstruction. The distance metric resulting from these weights can then be used to build other minimum spanning trees using other combinations of the remaining replicates of all cell types. We can then measure the consistency between each of the resulting trees and the correct differentiation tree and report on the robustness of our method. Further validation of the LP method can be done in a semi-supervised setting by learning a weighted distance metric from one part of the differentiation tree and then trying to reconstruct the whole tree using the same set of weights, or by learning a metric on the tree after removing one cell type and then trying to locate where that missing cell type should be using the learned metric. New

stages of differentiation can also be predicted by learning a weighted metric on a tree with known stages although that may require adjusting the upper limit on each constraint or even removing or adding certain constraints. The method can also be applied in a larger context by learning a distance metric on multiple differentiation trees for different types of cells such as blood cells, nerve cells, muscles cells, etc. This may allow us to find other sets of genes that are able to globally discriminate branches of differentiation for cells belonging to different body tissues.

Our choice of constraints was based on the assumption that the distance between two cell types A and B must be larger than the distance between any of those two cell types and another cell type C on the path connecting A and B in the tree. That does not only apply to cell types in the same lineage but even to cell types in different lineages, which will guarantee, by Theorem 4.1, that the differentiation tree is a minimum spanning tree of its cell types and therefore allow for successful reconstruction. However, this assumption partially neglects the fact that differentiation is a directed process (de-differentiation is rare) while distance metrics, by definition, are symmetric. We may choose to analyze the differentiation tree from a vertical perspective so that fully-differentiated cell types are considered more similar to each other than a fully-differentiated cell type and a stem cell type are. Similarly, a progenitor cell would be considered more similar to another progenitor cell in a different lineage than it is to a fully-differentiated cell in the same lineage. This new way of describing the tree will allow us to select genes that are not necessarily unique to certain subtrees but are unique to cell types at similar differentiation levels. Examples of such genes are those that regulate cell death and/or cell division and genes that trigger differentiation, pause it and/or terminate it. In this case, the assumption we made in Theorem 4.1 will still apply to cell types A, B and C if they all belong to the same lineage. However, when the three cell types are not all in one lineage, our initial assumption will apply in some cases but not in other cases depending on the differentiation levels of the three cells. Therefore, this new description of the tree that will identify other genes of interest can be accommodated in our method by adding new constraints to the linear program and removing some of the

present constraints. However, the new set of constraints may not be sufficient for reconstructing the differentiation tree as a minimum spanning tree.

Our LP approach could also be generalized to relate other genomic-wide signals with differentiation hierarchies. For example, if one has ChIP-Seq data measuring transcription factor binding or chromatin marks, one might be interested in identifying a small set of binding sites or marked sites that discriminate different types of cells. These could represent key regulatory events or types of regulation that help to determine stem cell identity. Our approach could also be used to relate gene expression data with the structures of phylogenetic trees, which may offer more insights into understanding the ancestral relationships between species on a genome-wide scale. Therefore, our methods and results can be expanded in different directions and can help answer many other research questions related to stem cell differentiation as well as other hierarchies of events.

REFERENCES

Aiba, K. et al (2006) Defining a developmental path to neural fate by global expression profiling of mouse embryonic stem cells and adult neural stem/progenitor cells. *Stem Cells*, 24 (4), 889-895.

Aiba, K. et al (2009) Defining developmental potency and cell lineage trajectories by expression profiling of differentiating mouse embryonic stem cells. *DNA Research*, 16 (1), 73-80.

Akashi, K. (2005) Lineage promiscuity and plasticity in hematopoietic development. *Annals of the New York Academy of Sciences*, 1044 (1), 125-131.

Amit, I. et al. (2009) Unbiased reconstruction of a mammalian transcriptional network mediating pathogen responses. *Science*, 326 (5950), 257-263.

Aplan, P. D. et al. (1992) The SCL gene product: a positive regulator of erythroid differentiation. *The EMBO Journal*, 11 (11), 4073-4081.

Ashburner, M. et al. (2000) Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25 (1), 25-29.

Bakker, W. J. et al. (2007) Differential regulation of Foxo3a target genes in erythropoiesis. *Molecular and Cellular Biology*, 27 (10), 3839-3854.

Beasley, J. E. & Beasley, J. E. (Eds) (1996) Advances in linear and integer programming. *Oxford: Clarendon Press*.

Berkelaar, M. et al. (2007) lpsolve: A mixed integer linear programming (MILP) solver. *URL http://sourceforge.net/projects/lpsolve.*

Boyer, L. A. et al. (2005) Core transcriptional regulatory circuitry in human embryonic stem cells. *Cell*, 122 (6), 947-956.

Breiman, L. (2001) Random forests. *Machine Learning*, 45 (1), 5-32.

Brunet, J. P. et al (2004) Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the National Academy of Sciences*, 101 (12), 4164-4169.

Davidson, E. H. (2001) Genomic regulatory systems: in development and evolution. *Academic Press*.

Felsenstein, J. (1981) Evolutionary trees from DNA sequences: a maximum likelihood approach. *Journal of Molecular Evolution*, 17 (6), 368-376.

Fitch, W. M. (1971) Toward defining the course of evolution: minimum change for a specific tree topology. *Systematic Biology*, 20 (4), 406-416.

Garey, M. R. & Johnson, D. S. (2002) Computers and intractability, Vol. 29, with freeman.

Globerson, A. & Roweis S. (2005) Metric learning by collapsing classes. *NIPS*, 18, 451-458.

Graham, R. L. & Hell, P. (1985) On the history of the minimum spanning tree problem. *Annals of the History of Computing*, 7 (1), 43-57.

Harlow, L. L., Mulaik, S. A. & Steiger, J. H., eds. (2013) What if there were no significance tests?. *Psychology Press*.

Hoffmann, R., Seidl, T. & Dugas, M. (2002) Profound effect of normalization on detection of differentially expressed genes in oligonucleotide microarray data analysis. *Genome Biol*, 3 (7), 0033.1-0033.11.

Huang, D. W. et al. (2009) Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nature Protocols*, 4 (1), 44-57.

Iwasaki, H. & Akashi, K. (2007) Myeloid lineage commitment from the hematopoietic stem cell. *Immunity*, 26 (6), 726-740.

Joshi, A. & Berthold G. (2011) Maximum parsimony analysis of gene expression profiles permits the reconstruction of developmental cell lineage trees. *Developmental Biology*, 353 (2), 440-447.

Kanehisa, M. & Goto, S. (2000) KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 28 (1), 27-30.

Kiel, M. J. et al. (2005) SLAM family receptors distinguish hematopoietic stem and progenitor cells and reveal endothelial niches for stem cells. *Cell*, 121 (7), 1109-1121.

Kim, S. Y., Lee, J. W. & Bae, J. S. (2006) Effect of data normalization on fuzzy clustering of DNA microarray data. *BMC Bioinformatics*, 7 (1), 134.

Kluger, Y. et al. (2004) Lineage specificity of gene expression patterns. *Proceedings of the National Academy of Sciences of the United States of America*, 101 (17), 6508-6513.

Kwok, J. T. & Tsang, I. W. (2003) Learning with idealized kernels. *ICML*, 400-407.

Krause, D. S. et al. (2001) Multi-organ, multi-lineage engraftment by a single bone marrow-derived stem cell. *Cell*, 105 (3), 369-377.

Lagasse, E. et al. (2000) Purified hematopoietic stem cells can differentiate to hepatocytes *in vivo*. *Nature Medicine*, 6 (11), 1229-1234.

Luo, F. et al (2004) A dynamically growing self-organizing tree (DGSOT) for hierarchical clustering gene expression profiles. *Bioinformatics*, 20 (16), 2605-2617.

Maki, N. et al. (2009) Expression of stem cell pluripotency factors during regeneration in newts. *Developmental Dynamics*, 238 (6), 1613-1616.

Matoba, R. et al (2006) Dissecting Oct3/4-regulated gene networks in embryonic stem cells by expression profiling. *PLoS One,* 1 (1), e26.

Matsuzaki, Y. et al. (2004) Unexpectedly efficient homing capacity of purified murine hematopoietic stem cells. *Immunity*, 20 (1), 87-93.

Morrison, S. J. & Spradling, A. C. (2008) Stem cells and niches: mechanisms that promote stem cell maintenance throughout life. *Cell*, 132 (4), 598-611.

Morrison, S. J. & Weissman, I. L. (1994) The long-term repopulating subset of hematopoietic stem cells is deterministic and isolatable by phenotype. *Immunity,* 1 (8), 661-673.

Morrison, S. J. et al. (1997) Identification of a lineage of multipotent hematopoietic progenitors. *Development,* 124 (10), 1929-1939.

Müller, F. J. et al (2008) Regulatory networks define phenotypic classes of human stem cell lines, *Nature*, 455 (7211), 401-405.

Murtagh, F. (1983) A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal*, 26 (4), 354-359.

Ng, S. Y., Yoshida, T. & Georgopoulos, K. (2007) Ikaros and chromatin regulation in early hematopoiesis. *Current Opinion in Immunology*, 19 (2), 116-122.

Novershtern, N. et al. (2011) Densely interconnected transcriptional circuits control cell states in human hematopoiesis. *Cell*, 144 (2), 296-309.

Papadimitriou, C. H. (1981) On the complexity of integer programming. *Journal of the ACM (JACM)*, 28 (4), 765-768.

Prim, R. C. (1957) Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36 (6), 1389-1401.

Quackenbush, J. (2001) Computational analysis of microarray data. *Nature Reviews Genetics*, 2 (6), 418-427.

Quesenberry, P. J. et al. (2005) Stem cell biology and the plasticity polemic. *Experimental Hematology*, 33 (4), 389-394.

Reya, T. et al. (2001) Stem cells, cancer, and cancer stem cells. *Nature*, 414 (6859), 105-111.

Rosenbauer, F. & Tenen, D. G. (2007) Transcription factors in myeloid development: balancing differentiation with transformation. *Nature Reviews Immunology*, 7 (2), 105-117.

Schmidt, F. L. (1996) Statistical significance testing and cumulative knowledge in psychology: Implications for training of researchers. *Psychological Methods*, 1 (2), 115-129.

Segal, E. et al. (2003) Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics*, 34 (2), 166-176.

Shanker, M., Hu, M. Y. &Hung, M. S. (1996) Effect of data standardization on neural network training. *Omega*, 24 (4), 385-397.

Sharov, A. A. et al. (2003) Transcriptome analysis of mouse stem cells and early embryos. *PLoS Biology*, 1 (3), e74.

Sharov, A. A., Dudekula, D. B. & Ko, M. S. (2005) A web-based tool for principal component and significance analysis of microarray data. *Bioinformatics*, 21 (10), 2548-2549.

Sola, J. & Sevilla, J. (1997) Importance of input data normalization for the application of neural networks to complex industrial problems. *Nuclear Science, IEEE Transactions on*, 44 (3), 1464-1468.

Spangrude, G. J., Heimfeld, S. & Weissman, I. L. (1988) Purification and characterization of mouse hematopoietic stem cells. *Science*, 241 (4861), 58-62.

Suzuki, H. et al (2009) The transcriptional network that controls growth arrest and differentiation in a human myeloid leukemia cell line. *Nature Genetics*, 41 (5), 553-562.

Takahashi, K. & Yamanaka, S. (2006) Induction of pluripotent stem cells from mouse embryonic and adult fibroblast cultures by defined factors. *Cell*, 126 (4), 663-676.

Takano, H. et al. (2004) Asymmetric division and lineage commitment at the level of hematopoietic stem cells inference from differentiation in daughter cell and granddaughter cell pairs. *The Journal of Experimental Medicine*, 199 (3), 295-302.

Weinberger, K. et al. (2006) Distance metric learning for large margin nearest neighbor classification. *Advances in Neural Information Processing Systems*, 18 (1473).

Weinberger, K. Q. & Saul, L. K. (2008) Fast solvers and efficient implementations for distance metric learning. *Proceedings of the 25th International Conference on Machine Learning*, 1160-1167, ACM.

Weissman, I. L. (1994) Stem cells, clonal progenitors, and commitment to the three lymphocyte lineages: T, B, and NK cells. *Immunity*, 1 (7), 529-531.

Xing, E.P. et al. (2003) Distance metric learning with application to clustering with side-information. *Advances in Neural Information Processing Systems*, 521-528.

Yang, L. & Jin, R. (2006) Distance metric learning: A comprehensive survey. *Michigan State University* 2.

Ying, Y. & Li, P. (2012) Distance metric learning with eigenvalue optimization. *The Journal of Machine Learning Research*, 13, 1-26.

Zola, H. et al. (2007) CD molecules2006---human cell differentiation molecules. *Journal of Immunological Methods*, 319 (1), 1-5.

APPENDIX

List of the 175 genes with positive weights in the 175-gene solution found by the LP approach

Gene Rank: Rank by weight in the 175-gene solution
Relative Weight: Weight relative to the maximum weight in the 175-gene solution
Actual Weight: Actual weight in the 175-gene solution
Gene Symbol: Gene symbol from UniGene
Gene Title: Title of a gene
Entrez Gene: Entrez gene database UID

Gene Rank	Relative Weight	Actual Weight	Gene Symbol	Gene Title	ID	Entrez Gene
1	1	0.396624	EPOR	erythropoietin receptor	215054_at	2057
2	0.651302	0.258322	ZNF198	zinc finger protein 198	210281_s_at	7750
3	0.602072	0.238796	CD1C	CD1C antigen, c polypeptide	205987_at	911
4	0.505699	0.200572	PHF14	PHD finger protein 14	216104_at	9678
5	0.501753	0.199007	RS1	retinoschisis (X-linked, juvenile) 1	216937_s_at	6247
6	0.479695	0.190259	CCR9	chemokine (C-C motif) receptor 9	207445_s_at	10803
7	0.465634	0.184682	LOC57228	small trans-membrane & glycosylated protein	209679_s_at	57228
8	0.442984	0.175698	CD24	CD24 antigen (small cell lung carcinoma cluster 4 antigen)	208651_x_at	934
9	0.429168	0.170218	DDX19A	DEAD (Asp-Glu-Ala-As) box polypeptide 19A	202578_s_at	55308
10	0.4242	0.168248	SULT2A1	sulfotransferase family, cytosolic, 2A, dehydroepiandrosterone (DHEA)-preferring, member 1	206292_s_at	6822
11	0.392001	0.155477	ACACB	acetyl-Coenzyme A carboxylase beta	214584_x_at	32
12	0.366683	0.145436	DKFZP564O0823	DKFZP564O0823 protein	204687_at	25849
13	0.366165	0.14523	IBTK	inhibitor of Bruton agammaglobulinemia tyrosine kinase	215086_at	25998
14	0.35957	0.142614	RFP	ret finger protein	210541_s_at	5987

15	0.35768	0.141865	COL4A3	collagen, type IV, alpha 3 (Goodpasture antigen)	222073_at	1285
16	0.354526	0.140614	TCN1	transcobalamin I (vitamin B12 binding protein, R binder family)	205513_at	6947
17	0.34932	0.138549	SLC7A2	solute carrier family 7 (cationic amino acid transporter, y+ system), member 2	207626_s_at	6542
18	0.346191	0.137308	HERC5	hect domain and RLD 5	219863_at	51191
19	0.343563	0.136265	GNRHR	gonadotropin-releasing hormone receptor	211522_s_at	2798
20	0.336362	0.133409	DUS2L	dihydrouridine synthase 2-like (SMM1, S. cerevisiae)	47105_at	54920
21	0.334321	0.1326		Transcribed locus, weakly similar to XP_518244.1 PREDICTED: similar to sirtuin 5 isoform 2; sir2-like 5; silent mating type information regulation 2, S.cerevisiae, homolog 5;	222315_at	
22	0.333092	0.132113	EHD3	EH-domain containing 3	218935_at	30845
23	0.330924	0.131253	ESR2	estrogen receptor 2 (ER beta)	211120_x_at	2100
24	0.324581	0.128737	P2RX4	purinergic receptor P2X, ligand-gated ion channel, 4	204088_at	5025
25	0.311339	0.123485	IGSF3	immunoglobulin superfamily, member 3	202421_at	3321
26	0.306991	0.12176	HLXB9	homeo box HB9	214614_at	3110
27	0.265365	0.10525	LAG3	lymphocyte-activation gene 3	206486_at	3902
28	0.26535	0.105244	CAMSAP 1	calmodulin regulated spectrin-associated protein 1	212712_at	157922
29	0.265006	0.105108	MLH3	mutL homolog 3 (E. coli)	214525_x_at	27030
30	0.262318	0.104042	ATPAF2	ATP synthase mitochondrial F1 complex assembly factor 2	214330_at	91647
31	0.251372	0.0997	OGFR	opioid growth factor receptor	211513_s_at	11054
32	0.242678	0.096252	NFE2L1	nuclear factor (erythroid-derived 2)-like 1	200758_s_at	4779
33	0.242205	0.096064	HFE	hemochromatosis	211332_x_at	3077
34	0.241839	0.095919	DPP6	dipeptidylpeptidase 6	207789_s_at	1804
35	0.237549	0.094218	TBX1	T-box 1	207662_at	6899

36	0.231562	0.091843	APOC1	apolipoprotein C-I	213553_x_at	341
37	0.231446	0.091797	MGC11061	Yip1 domain family, member 4	213999_at	84272
38	0.230392	0.091379		CDNA FLJ12815 fis, clone NT2RP2002546	213169_at	
39	0.230109	0.091267	HLA-DQB1	major histocompatibility complex, class II, DQ beta 1	210747_at	3119
40	0.222764	0.088354	C1orf61	chromosome 1 open reading frame 61	205103_at	10485
41	0.222187	0.088125	FGF2	fibroblast growth factor 2 (basic)	204422_s_at	2247
42	0.219441	0.087036	CYC1	cytochrome c-1	201066_at	1537
43	0.213932	0.084851	CD8B1	CD8 antigen, beta polypeptide 1 (p37)	207979_s_at	926
44	0.213646	0.084737	EDA	ectodysplasin A	211129_x_at	1896
45	0.212489	0.084278	OLR1	oxidised low density lipoprotein (lectin-like) receptor 1	210004_at	4973
46	0.211913	0.08405	MMPL1	matrix metallopeptidase-like 1	207289_at	4328
47	0.210052	0.083312	CTSW	cathepsin W (lymphopain)	214450_at	1521
48	0.206349	0.081843	CDC42EP2	CDC42 effector protein (Rho GTPase binding) 2	209850_s_at	10435
49	0.204101	0.080951	TIAM1	T-cell lymphoma invasion and metastasis 1	213135_at	7074
50	0.20243	0.080289	TRFP	Trf (TATA binding protein-related factor)-proximal homolog (Drosophila)	206961_s_at	9477
51	0.197153	0.078196	CD40LG	CD40 ligand (TNF superfamily, member 5, hyper-IgM syndrome)	207892_at	959
52	0.196691	0.078012	THOC2	THO complex 2	212994_at	57187
53	0.195342	0.077478	CDK9	cyclin-dependent kinase 9 (CDC2-related kinase)	203198_at	1025
54	0.195322	0.077469	VN1R1	vomeronasal 1 receptor 1	221412_at	57191
55	0.193766	0.076852	FLJ10661	hypothetical protein FLJ10661	220353_at	55199
56	0.193248	0.076647	IGFBP4	insulin-like growth factor binding protein 4	201508_at	3487
57	0.191692	0.07603	CLU	clusterin (complement lysis inhibitor, SP-40,40, sulfated glycoprotein 2, testosterone-repressed prostate message 2, apolipoprotein J)	208791_at	1191

58	0.186496	0.073969	NOD9	NOD9 protein	219680_at	79671
59	0.184537	0.073192	C7orf16	chromosome 7 open reading frame 16	220231_at	10842
60	0.184424	0.073147	CD8A	CD8 antigen, alpha polypeptide (p32)	205758_at	925
61	0.184034	0.072993	PXMP2	peroxisomal membrane protein 2, 22kDa	219076_s_at	5827
62	0.180691	0.071667	MLANA	melan-A	206427_s_at	2315
63	0.17817	0.070666	FANCA	Fanconi anemia, complementation group A	203805_s_at	2175
64	0.174379	0.069163	PML /// LOC161527	promyelocytic leukemia /// hypothetical protein LOC161527	211014_s_at	
65	0.171103	0.067864	OR12D2	olfactory receptor, family 12, subfamily D, member 2	221344_at	26529
66	0.169753	0.067328	IL1A	interleukin 1, alpha	208200_at	3552
67	0.15571	0.061759	PDE8B	phosphodiesterase 8B	213228_at	8622
68	0.153421	0.06085	FLJ23554	hypothetical protein FLJ23554	220141_at	79864
69	0.153373	0.060831	SAPS2	SAPS domain family, member 2	202791_s_at	9701
70	0.146208	0.05799	NPY2R	neuropeptide Y receptor Y2	210730_s_at	4887
71	0.137732	0.054628	ADORA1	adenosine A1 receptor	205481_at	134
72	0.134544	0.053364	RDH11	retinol dehydrogenase 11 (all-trans and 9-cis)	217775_s_at	51109
73	0.131793	0.052272	BNC2	basonuclin 2	220272_at	54796
74	0.129492	0.05136	MBTPS2	membrane-bound transcription factor peptidase, site 2	206473_at	51360
75	0.128843	0.051102	C14orf56	chromosome 14 open reading frame 56	215877_at	89919
76	0.128375	0.050916	ISL1	ISL1 transcription factor, (islet-1),LIM/homeodomain	206104_at	3670
77	0.127488	0.050565	MGC4692	hypothetical protein MGC4692	218410_s_at	79118
78	0.125359	0.04972			208187_s_at	
79	0.125047	0.049597	RPS6KA2	ribosomal protein S6 kinase, 90kDa, polypeptide 2	212912_at	6196
80	0.123286	0.048898	OR1D4 /// OR1D5	olfactory receptor, family 1, subfamily D, member 4 /// olfactory receptor, family 1, subfamily D, member 5	221341_s_at	
81	0.122457	0.048569	DNM1	dynamin 1	215116_s_at	1759

82	0.122123	0.048437	MAPK7	mitogen-activated protein kinase 7	35617_at	5598
83	0.121359	0.048134	FLJ13236	hypothetical protein FLJ13236	220441_at	79962
84	0.115254	0.045712	MYO5C	Myosin VC	215229_at	55930
85	0.11509	0.045647	KCNN4	potassium intermediate/ small conductance calcium-activated channel, subfamily N, member 4	204401_at	3783
86	0.113879	0.045167	PIK3R3	phosphoinositide-3-kinase, regulatory subunit 3 (p55, gamma)	211580_s_at	8503
87	0.113264	0.044923	FLJ22795 /// LOC388152 /// LOC388161	hypothetical protein FLJ22795 /// hypothetical protein FLJ90297 /// LOC388161	220602_s_at	
88	0.110594	0.043864	PDGFB	platelet-derived growth factor beta polypeptide (simian sarcoma viral (v-sis) oncogene homolog)	204200_s_at	5155
89	0.104319	0.041375			217107_at	
90	0.102956	0.040835		Similar to Peptidyl-prolyl cis-trans isomerase E (PPIase E) (Rotamase E) (Cyclophilin E) (Cyclophilin 33)	222054_at	440582
91	0.102539	0.040669	SMU1	smu-1 suppressor of mec-8 and unc-52 homolog (C. elegans)	218393_s_at	55234
92	0.102149	0.040515	GYS1	glycogen synthase 1 (muscle)	201673_s_at	2997
93	0.09991	0.039627	KIAA0676	KIAA0676 protein	206431_x_at	23061
94	0.09852	0.039075	TAOK2	TAO kinase 2	204877_s_at	9344
95	0.094402	0.037442	CD44	CD44 antigen (homing function and Indian blood group system)	204490_s_at	960
96	0.093665	0.03715	T	T, brachyury homolog (mouse)	206524_at	6862
97	0.093612	0.037129	PTPN13	protein tyrosine phosphatase, non-receptor type 13 (APO1/CD95 (Fas)-associated phosphatase)	204201_s_at	5783

98	0.09264	0.036743		Clone 24405 mRNA sequence	213832_at	
99	0.092607	0.03673	HSPBP1	hsp70-interacting protein	202415_s_at	23640
100	0.092367	0.036635	GABARAPL1 /// GABARAPL3	GABA(A) receptor-associated protein like 1 /// GABA(A) receptors associated protein like 3	211458_s_at	
101	0.08698	0.034498	NDUFB7	NADH dehydrogenase (ubiquinone) 1 beta subcomplex, 7, 18kDa	211407_at	4713
102	0.083377	0.033069	ADRM1	adhesion regulating molecule 1	201281_at	11047
103	0.083305	0.033041	GEM	GTP binding protein overexpressed in skeletal muscle	204472_at	2669
104	0.082725	0.032811	LOC257407	hypothetical protein LOC257407	213148_at	257407
105	0.081661	0.032389	MAFF	v-maf musculoaponeurotic fibrosarcoma oncogene homolog F (avian)	205193_at	23764
106	0.077911	0.030901	LTF	Lactotransferrin	202018_s_at	4057
107	0.076548	0.030361			213690_s_at	
108	0.075394	0.029903	LY6G5C	lymphocyte antigen 6 complex, locus G5C	219860_at	80741
109	0.075171	0.029814	PLA2G2A	phospholipase A2, group IIA (platelets, synovial fluid)	203649_s_at	5320
110	0.075023	0.029756	IFI35	interferon-induced protein 35	209417_s_at	3430
111	0.074771	0.029656	RPS6KA6	ribosomal protein S6 kinase, 90kDa, polypeptide 6	220737_at	27330
112	0.072485	0.028749	ATP6V1C1	ATPase, H+ transporting, lysosomal 42kDa, V1 subunit C, isoform 1	202872_at	528
113	0.07233	0.028688	HRH4	histamine receptor H4	221169_s_at	59340
114	0.071147	0.028219	OLFML2A	olfactomedin-like 2A	213075_at	169611
115	0.070482	0.027955	RECK	reversion-inducing-cysteine-rich protein with kazal motifs	216156_at	8434
116	0.070405	0.027924	TM9SF1	transmembrane 9 superfamily member 1	209149_s_at	10548
117	0.070023	0.027773		Similar to Zinc finger protein 492	217544_at	388760
118	0.068677	0.027239	CCR3	chemokine (C-C motif) receptor 3	208304_at	1232

119	0.066934	0.026548	MAPK8IP 3	mitogen-activated protein kinase 8 interacting protein 3	213178_s_at	23162
120	0.062419	0.024757	TRIM22	tripartite motif-containing 22	213293_s_at	10346
121	0.061273	0.024302	RFX5	regulatory factor X, 5 (influences HLA class II expression)	202964_s_at	5993
122	0.054006	0.02142	ESR2	estrogen receptor 2 (ER beta)	211117_x_at	2100
123	0.053478	0.021211	RNF40	ring finger protein 40	217642_at	9810
124	0.051959	0.020608	COL4A3	collagen, type IV, alpha 3 (Goodpasture antigen)	214641_at	1285
125	0.051168	0.020295	EIF4G3	Eukaryotic translation initiation factor 4 gamma, 3	216146_at	8672
126	0.050105	0.019873	NNT	Nicotinamide nucleotide transhydrogenase	215278_at	23530
127	0.048786	0.01935	PCDH21	protocadherin 21	213369_at	92211
128	0.048579	0.019268	CD177	CD177 antigen	219669_at	57126
129	0.047866	0.018985	PML /// LOC1615 27	promyelocytic leukemia /// hypothetical protein LOC161527	211012_s_at	
130	0.047496	0.018838	KIAA068 3	KIAA0683 gene product	34260_at	9894
131	0.047448	0.018819	PLOD3	procollagen-lysine, 2-oxoglutarate 5-dioxygenase 3	202185_at	8985
132	0.047261	0.018745	MAN2A2	mannosidase, alpha, class 2A, member 2	219999_at	4122
133	0.045953	0.018226	TLR3	toll-like receptor 3	206271_at	7098
134	0.045165	0.017913	HPS6	Hermansky-Pudlak syndrome 6	219052_at	79803
135	0.043235	0.017148	RNF24	Ring finger protein 24	216179_x_at	11237
136	0.043095	0.017092	KLHL20	kelch-like 20 (Drosophila)	210634_at	27252
137	0.041747	0.016558	IL12A	interleukin 12A (natural killer cell stimulatory factor 1, cytotoxic lymphocyte maturation factor 1, p35)	207160_at	3592
138	0.038879	0.01542	KIAA021 7	La ribonucleoprotein domain family, member 5	214215_s_at	23185
139	0.034921	0.013851	GPNMB	glycoprotein (transmembrane)	201141_at	10457
140	0.034492	0.01368	ACVR1	activin A receptor, type I	203935_at	90
141	0.033919	0.013453	CCHCR1	coiled-coil alpha-helical rod protein 1	209698_at	54535

142	0.029474	0.01169	THRB	thyroid hormone receptor, beta (erythroblastic leukemia viral (v-erb-a) oncogene homolog 2, avian)	207044_at	7068
143	0.028509	0.011307			217172_at	
144	0.027689	0.010982	SPAG9	sperm associated antigen 9	206748_s_at	9043
145	0.026739	0.010605	SOX5	SRY (sex determining region Y)-box 5	215868_x_at	6660
146	0.026593	0.010547	FAM13A 1	family with sequence similarity 13, member A1	202973_x_at	10144
147	0.025859	0.010256	CSNK1G1	casein kinase 1, gamma 1	220640_at	53944
148	0.024467	0.009704	EVI1	ecotropic viral integration site 1	215851_at	2122
149	0.023902	0.00948	MSH5	mutS homolog 5 (E. coli)	210410_s_at	4439
150	0.023304	0.009243	CACNA1 G	calcium channel, voltage-dependent, alpha 1G subunit	207869_s_at	8913
151	0.022651	0.008984	WASL	Wiskott-Aldrich syndrome-like	205809_s_at	8976
152	0.022043	0.008743	ANKRD6	ankyrin repeat domain 6	204671_s_at	22881
153	0.0192	0.007615	ACLY	ATP citrate lyase	210337_s_at	47
154	0.018902	0.007497	UFD1L	ubiquitin fusion degradation 1 like (yeast)	209103_s_at	7353
155	0.018558	0.00736	MCP	membrane cofactor protein (CD46, trophoblast-lymphocyte cross-reactive antigen)	211574_s_at	4179
156	0.018371	0.007286	SDC2	syndecan 2 (heparan sulfate proteoglycan 1, cell surface-associated, fibroglycan)	212158_at	6383
157	0.01762	0.006988	PURA	Purine-rich element binding protein A	213806_at	5813
158	0.015768	0.006254	DCTD	dCMP deaminase	201571_s_at	1635
159	0.014609	0.005794	F5	coagulation factor V (proaccelerin, labile factor)	204713_s_at	2153
160	0.01453	0.005763	PRKCZ	protein kinase C, zeta	202178_at	5590
161	0.014357	0.005694	LOC2840 01	hypothetical protein LOC284001	214818_at	284001
162	0.012678	0.005029	SURF2	surfeit 2	205224_at	6835
163	0.012569	0.004985	DKFZp56 6H0824	hypothetical protein DKFZp566H0824	207470_at	54744
164	0.012105	0.004801	IL3RA	interleukin 3 receptor, alpha (low affinity)	206148_at	3563

165	0.011836	0.004695	ATP10A	ATPase, Class V, type 10A	214256_at	57194
166	0.011606	0.004603	TRD@	T cell receptor delta locus	213830_at	6964
167	0.011334	0.004495	OBSCN	obscurin, cytoskeletal calmodulin and titin-interacting RhoGEF	220427_at	84033
168	0.008862	0.003515	KRT5	keratin 5 (epidermolysis bullosa simplex, Dowling-Meara/Kobner/Weber-Cockayne types)	201820_at	3852
169	0.007785	0.003088	SLC29A3	solute carrier family 29 (nucleoside transporters), member 3	219344_at	55315
170	0.007136	0.00283	SSX3	synovial sarcoma, X breakpoint 3	211731_x_at	10214
171	0.006984	0.00277	PGBD5	piggyBac transposable element derived 5	219225_at	79605
172	0.006402	0.002539	EPHA2	EPH receptor A2	203499_at	1969
173	0.004681	0.001856	NIT1	nitrilase 1	202891_at	4817
174	0.004122	0.001635	SDF4	stromal cell derived factor 4	217855_x_at	51150
175	0.001006	0.000399	HIGD1A	HIG1 domain family, member 1A	217845_x_at	25994