Bioinformatics, 31(16), 2015, 2676–2682
doi: 10.1093/bioinformatics/btv192
Advance Access Publication Date: 5 April 2015
Original Paper

OXFORD

Gene expression

Gene selection for the reconstruction of stem cell differentiation trees: a linear programming approach

Mohamed A. Ghadie[1,2], Nathalie Japkowicz[1] and Theodore J. Perkins[1,2,3,*]

[1]School of Electrical Engineering and Computer Science, University of Ottawa, 75 Laurier Avenue East, Ottawa, ON K1N 6N5, Canada, [2]Ottawa Hospital Research Institute, 501 Smyth Road, Ottawa, Ontario K1H 8L6, Canada and [3]Department of Biochemistry, Microbiology and Immunology, University of Ottawa, 451 Smyth Road, Ottawa, ON K1H 8M5, Canada

*To whom correspondence should be addressed.

Associate Editor: Ziv Bar-Joseph

Received on July 29, 2014; revised on February 15, 2015; accepted on March 30, 2015

Abstract

Motivation: Stem cell differentiation is largely guided by master transcriptional regulators, but it also depends on the expression of other types of genes, such as cell cycle genes, signaling genes, metabolic genes, trafficking genes, etc. 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 can organize cell types into a tree, but in general this tree is different from the differentiation hierarchy itself.

Methods: Given the differentiation hierarchy and gene expression data at each node, we 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 approach to identify sparse sets of weights, effectively identifying genes that are most relevant for discriminating different parts of the tree.

Results: We apply our method to microarray gene expression data describing 38 cell types in the hematopoiesis hierarchy, constructing a weighted Euclidean metric that uses just 175 genes. 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. We then report on the selected genes and their biological functions. Our approach offers a new way to identify genes that may have important roles in stem cell differentiation.

Contact: tperkins@ohri.ca

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

The differentiation of stem cells from a pluripotent state towards increasingly specialized cell types has long been conceptualized as a branching process, with multi-potent cells towards the top and fully differentiated cells at the bottom layer (Reya *et al.*, 2001). The discrete cell types in the differentiation hierarchy have been identified by scientists using a number of means, including morphology, the study of development, lineage tracing, etc. In practice, the cell types

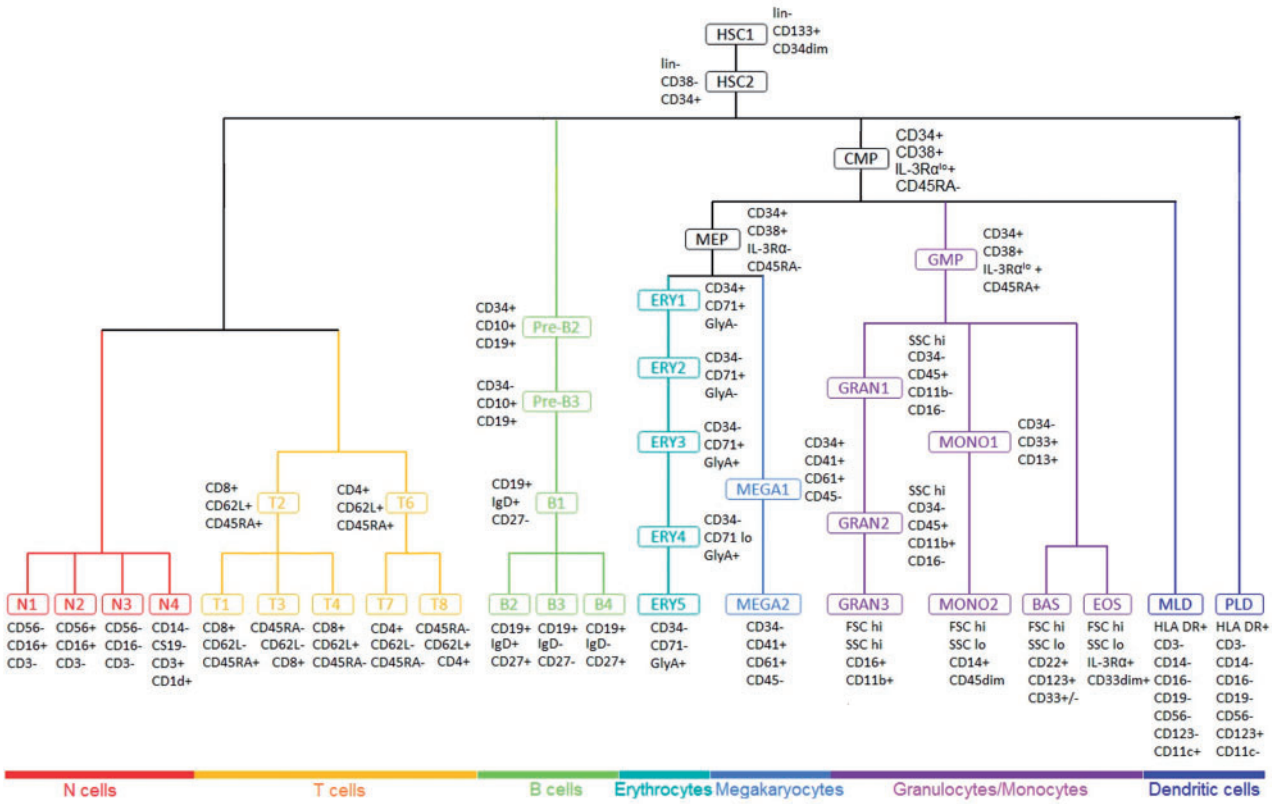

Fig. 1. Differentiation hierarchy for human blood cells as described in Novershtern *et al.* (2011). The 38 cell types are: hematopoietic stem cells (HSC1-2), common myeloid progenitor (CMP), megakaryocyte/erythroid progenitor (MEP), erythroid cells (ERY1-5), CFU-MK (MEGA1), megakaryocyte (MEGA2), granulocyte/monocyte progenitor (GMP), CGU-G (GRAN1), neutrophilic metamyelocyte (GRAN2), neutrophil (GRAN3), CFU-M (MONO1), monocyte (MONO2), eosinophil (EOS), basophil (BAS), myeloid dendritic cell (MLD), plasmacytoid dendritic cell (PLD), early B cell (Pre-B2), pre-B cell (Pre-B3), naïve B cell (B1), mature B cell class able to switch (B2), mature B cell (B3), mature B cell class switched (B4), mature NK cells (N1-4), naïve CD8+T cell (T2), CD8+ effector memory RA (T1), CD8+ effector memory (T3), CD8+ central memory (T4), naïve 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. A list of all marker genes used for cell type identification is shown in Supplementary Table S1 (Color version of this figure is available at *Bioinformatics* online.)

are often identified based on the expression of certain marker genes. Many of these are cell-surface proteins, which are fairly easy to assay biochemically (Morrison and Spradling, 2008). However, such marker genes 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. Since the advent of gene expression microarrays, and more recently RNA-Seq technology, expression in various stem cell types has been studied. This has prompted some authors to try to reconstruct differentiation hierarchies *de novo*, based only on expression data. Kluger *et al.* (2004) used single linkage hierarchical clustering on microarray gene expression data of mature, differentiated human blood cells to classify them into nine distinct lineages. 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 phylogenetic tree reconstruction, cellular differentiation hierarchies of mammalian tissue, specifically the hierarchical structures for hematopoietic differentiation, neural differentiation and early endoderm organogenesis, can be inferred from gene expression profiles by parsimonious maximization, again confirming the feasibility of *de novo* reconstruction. Principal component analysis (PCA) has also been used to map expression profiles of cells onto the principal component space of lower dimension (Aiba *et al.*, 2006, 2009). The positions of cells on the trajectories formed in the space of the highest three components seem to reflect the developmental potency of the cells and their differentiation status.

We became interested in testing reconstruction on the more extensive dataset of Novershtern *et al.* (2011). This data comprises microarray gene expression data for 22 215 genes in 38 distinct cell types in the hematopoiesis hierarchy (Fig. 1), each assayed with between 4 and 10 replicate arrays, for a total of 211 arrays (Supplementary Table S1). The majority of the cell types were purified with >95% purity from umbilical cord blood, which is enriched in undifferentiated populations. However, the terminally differentiated populations were purified from adult peripheral blood because exposure to antigens after birth is necessary for these populations. The remaining cells were identified by flow cytometry for labeled antibodies against the marker genes listed beside each cell type in Figure 1 and/or flow scatter properties. All marker genes that were used for cell type identification are also shown in Supplementary Table S1. [More details on cell populations and their sorting can be found in the experimental procedures of Novershtern *et al.* (2011)].

The hierarchical clustering approach espoused by Kluger *et al.* (2004) and the maximum parsimony approach of Joshi and Berthold (2011) assume data only at the most-differentiated level. Thus, they are not well-suited to the Novershtern data, which includes expression data at both leaves and internal nodes of the differentiation hierarchy. Nevertheless, by taking the unpalatable step

of discarding the 18 non-fully differentiated cell types, leaving only the 20 fully differentiated types, we were able to apply both approaches. The resulting trees bore substantial resemblance to the correct differentiation hierarchy, although there were some errors (Supplementary Figs S1 and S2).

It occurred to us that we might be able to reconstruct the differentiation hierarchy as a minimum spanning tree, because this would allow cell types to be either internal or leaf nodes in the resulting tree. We thus tried constructing minimum spanning trees on the expression data using various distance metrics (Euclidean, L1, cosine, correlation, Chebyshev, etc.). Accuracy of these reconstructions varied, but none of the trees matched the true differentiation tree (Supplementary Figs S3–S6). By performing PCA on the expression data of the 38 hematopoietic cell types, we were able to map the expression profiles of the cell types onto the space of the 37 principal components. The positions of the cells showed partial agreement with the correct hierarchy (Supplementary Fig. S7); however, again, when we constructed a minimum spanning tree of the 38 cells using Euclidean distance on the cell profiles in the new space of the n-highest components for $n = 1, \ldots, 37$, the resulting trees did not agree with the correct differentiation tree. Minimum spanning trees constructed using single principal components also did not agree with the differentiation hierarchy (Supplementary Fig. S8). We concluded that to reconstruct a differentiation hierarchy as a minimum spanning tree, we need to identify the correct distance metric.

Distance metric learning has been studied extensively in the machine learning literature, particularly in the context of classification. 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 and Tsang (2003) and Globerson and Roweis (2005). Weinberger *et al.* (2006), Weinberger and Saul (2008) and Ying and Li (2012) aim to improve K-nearest neighbor classification by learning a Mahalanobis distance metric from labeled training examples. A comprehensive survey of distance metric learning algorithms can also be found in Yang and Jin (2006). However, all of these methods focus on improving performance in classification tasks, usually binary classification. None of these methods is suitable to find a metric consistent with a differentiation hierarchy.

The main contribution of the present article is to propose a 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 propose a set of linear constraints on the weights, depending on the tree T and feature vectors, and prove that they are sufficient (though not necessary) to establish the desired distance metric. We then propose an efficiently solvable linear program designed to satisfy those 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. We apply this approach to the Novershtern data, demonstrating successful tree reconstruction, and report on genes selected by the optimization.

2 Materials and methods

2.1 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 2.1: 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

$$D(i,j) < D(i,k) \tag{1}$$

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 from N but not in M such that $D(u,v)$ is minimal and it 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 v in T. Also, the path from u to any node $z \neq v$ in M passes through v and by assumption, $D(u,v) < D(u,z)$. 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 iteration, M is a subtree of T, and at the end of Prim's algorithm, the two must be the same.

2.2 Finding a weighted Euclidean metric via linear programming

Theorem 2.1 gives us a set of conditions on a pairwise distance metric so that the minimum spanning tree matches a given tree. Here, we describe how to translate those constraints into a set of linear constraints on the weights of a weighted Euclidean distance metric. Additionally, we propose a linear programming approach to satisfying those constraints while also seeking a sparse solution. 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 $N = \{1, 2, \ldots, n\}$ where each node i is associated with an m-dimensional expression vector $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,m})$.

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 $x_i, x_j \in X$

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

Then using this weighted distance metric, (1) can be written as follows after squaring both sides of the inequality

$$\sum_{c=1}^{m}(x_{i,c}-x_{j,c})^2 w_c < \sum_{c=1}^{m}(x_{i,c}-x_{k,c})^2 w_c \quad (3)$$

Moving all terms to the left side, we rewrite (3) in the following form

$$\sum_{c=1}^{m}\left[(x_{i,c}-x_{j,c})^2-(x_{i,c}-x_{k,c})^2\right]w_c < 0 \quad (4)$$

To make sure the condition in (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 be true for the cases when k is an immediate neighbor of j since, by transitive law, satisfying (1) for i, k and its immediate neighbors 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 the inequality in (4) for each node i in T. These constraints can be grouped into the following matrix form

$$\mathbf{A}_i\mathbf{w} < 0 \quad (5)$$

If there is any vector of weights $\mathbf{w}$ satisfying these constraints, they will provide a weighted Euclidean metric that will, by Theorem 2.1, make the given tree T a minimum spanning tree. Those weights $\mathbf{w}$, however, may not uniquely solve the set of constraints. Following a common heuristic in distance metric learning, we propose to favor 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:

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

Where $\mathbf{A}=\begin{bmatrix}\mathbf{A}_1\\ \vdots \\ \mathbf{A}_n\end{bmatrix}$ and where $\mathbf{b}=\begin{bmatrix}b_1\\ \vdots \\ b_m\end{bmatrix}$ can be chosen arbitrarily, as long as $b_1,\ldots,b_m<0$. It can be shown that the number of rows/constraints in $\mathbf{A}$ scales as the square of the number of nodes n and is exactly equal to $(n-1)(n-2)$. We make the proof to this available as part of the supplementary material. 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 (unweighted) 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. If we solve this as an ordinary linear program (where weights are real-valued), solutions are known to favor a limited number of non-zero 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.

2.3 Constructing metrics with random subsets of genes

Minimizing the sum of weights is a heuristic that favors sparse solutions. However, there may be multiple solutions with the same sum of weights, and there may be other solutions with larger weight sums which are still 'good' solutions in some sense (such as having a smaller number of non-zero weights). Therefore, to better evaluate the importance of each gene in the Novershtern data, we explored an approach similar to that used in Random Forests (Breiman, 2001). We solve 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 choose a sample size that is small enough to allow for diversity among the samples and large enough to generate a feasible linear program. Since not all 100 programs will necessarily be feasible (i.e. a vector that satisfies all constraints simultaneously may not exist for each program) we calculate the score S_i for each gene i by integrating its ranks from all feasible solutions using the following equation:

$$S_i=\sum_{k=1}^{F}\left(1-\frac{r_{ik}-1}{W_k}\right) \quad (6)$$

Where F is the number of feasible programs, W_k is the number of non-zero weights in the k^{th} solution, and r_{ik} is the rank of gene i among the genes with non-zero weights in the k^{th} solution. The rank of a gene in a particular solution is included in its score calculation only if it receives a non-zero weight in that solution. By examining the scores of all genes we can see which important genes were left out of the metric we learned in section 2.2 and how important are the genes that were included in the metric.

3 Results

Our differentiation tree has $n=38$ nodes and each replicate array in the dataset consists of 22 215 gene expression values. Therefore, the matrix $\mathbf{A}$ in our linear program has $37\times 36=1332$ constraints/rows and 22 215 columns. For the purpose of computational accuracy and to be fair with all genes, we averaged the expression of each gene across all replicates of a cell so that each cell is represented by one array. We then normalized the data so that the mean expression of each gene over all cells is 0 with a standard deviation of 1. For the inequalities in our linear program, we chose $\mathbf{b}=-1^T$. Any other negative constant produces the same results, although attaching different constants to different constraints could produce different results.

3.1 A weighted Euclidean metric on 175 genes allows reconstruction of the differentiation tree

We tried finding a weighted Euclidean metric that could reproduce the correct differentiation tree by solving the linear program described in Section 2.2, using the program lp_solve (Berkelaar *et al.* 2007; http://sourceforge.net/projects/lpsolve). We found a solution with 175 genes receiving a non-zero weight and the remaining 22 040 genes receiving zero weight. A full list and description of the 175 non-zero-weight genes is available as an Excel spreadsheet as part of the supplementary material. The 175 genes bore little relationship to the 30 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 2 shows a heatmap of the 66 largest-weight genes, whereas Supplementary Figure S9 shows a heatmap of all 175. (Why we only show 66 genes here is explained shortly.) Most weighted genes are expressed primarily in one or a few lineages. For example, the largest weight goes to the

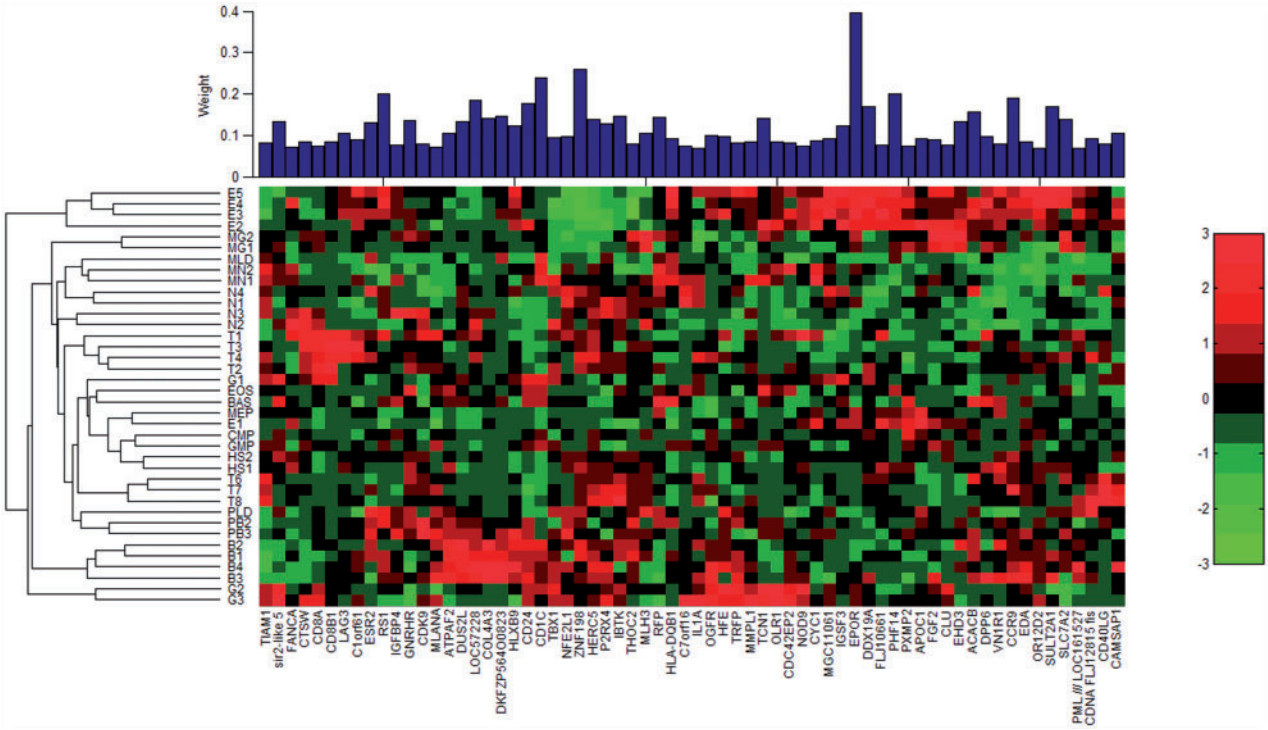

Fig. 2. Expression of genes with the largest 66 weights in the 175-gene solution. For visualization purposes, hierarchical clustering with the average linkage method was performed on the rows and columns using Euclidean distance. Thus, the dendrogram at left is not the same as the differentiation hierarchy, even though the minimum spanning tree based on the learning metric does match the differentiation hierarchy (Color version of this figure is available at *Bioinformatics* online.)

Erythropoietin receptor gene (EpoR), which is primarily expressed along the erythroid lineage, where it plays an important signaling role. The next largest weight goes to ZNF198, which is notably absent in erythroid cells, but present primarily in T- and B-cell lineages. Although the heatmap in Figure 2 is intended to show the un-weighted expression of the 66 largest-weight genes across different lineages, we do not expect the dendrogram to be perfectly consistent with the differentiation tree of Figure 1 since it was produced by hierarchical clustering. The early erythrocyte E1 was not clustered with the later erythrocytes but was clustered with its megakaryocyte/erythroid progenitor and the early stem cells, thus showing consistency with the results of Novershtern *et al.* (2011), where it is shown that E1 tends to associate itself with the early stem and progenitors based on regulation of transcription factor expression. The T cells T6-8 did not cluster with the other T cells in this heatmap; however, they do cluster together in the heatmap of all 175 genes shown in Supplementary Figure S9.

To investigate more whether the 175 genes selected by the linear program have any biological significance, we performed a DAVID functional annotation analysis (Huang *et al.*, 2009). The full results are available as an Excel spreadsheet as part of the supplementary material, but we highlight a few results here. The Gene Ontology (Ashburner *et al.*, 2000) biological process terms that showed the greatest enrichment were 'defense response' ($P = 3.22 \times 10^{-9}$), 'inflammatory response' ($P = 4.68 \times 10^{-7}$), 'response to wounding' ($P = 6.27 \times 10^{-7}$) and 'immune response' ($P = 2.16 \times 10^{-6}$)—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' ($P = 1.29 \times 10^{-3}$) and 'negative regulation of cell proliferation' ($P = 2.39 \times 10^{-3}$); these have obvious relevance to both intermediate cell types, where proliferation is typically ongoing, and differentiated cell types, which are mitotically quiescent. Many subsequent terms involve protein phosphorylation, no doubt reflecting the importance of cell signaling pathways in mediating differentiation decisions. The most significant KEGG (Kanehisa and Goto, 2000) pathway was 'hematopoietic cell lineage' ($P = 7.92 \times 10^{-4}$), followed by several cell signaling 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.

3.2 Reducing the number of genes in the distance metric

The number of genes selected by the learned metric is impressively small given the large number of genes in our dataset. Indeed, the chance of 175 random genes being able to reconstruct the whole tree is practically zero (Supplementary Fig. S10). However, 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 (22 215) 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 30 established marker genes shown in Figure 1. This problem turned out to be infeasible. We conjecture that non-linear 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 $\mathbf{Aw} \leq -1^T$, but theoretically we only need $\mathbf{Aw} < 0$ to ensure reconstruction of the tree. For $n = 1, 2, 3 \ldots$, 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. 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 2. It remains to be seen whether any smaller solution can be found.

3.3 Most of the 175 genes received large weights in randomly constructed metrics

To further evaluate the important of individual genes, we tried to construct 100 metrics each with a random set of 7000 genes. Out of the 100 attempts, 70 were successful, with the remainder generating infeasible linear programs. We then calculated the score of each gene over the 70 solutions as described in Section 2.3. Out of the 22 215 genes, 1573 received a non-zero weight in at least one solution and therefore received a non-zero score. A full list and description of these 1573 genes is available as an Excel spreadsheet as part of the supplementary material. We found a positive correlation between the average weight of each gene over all 70 solutions and the number of times it received a non-zero weight (Supplementary Fig. S11). This correlation was also evident in gene scores and their corresponding weights in the 175-gene solution (Fig. 3). All genes in the 175-gene group received non-zero scores and most of them scored in the top ranks, confirming their significance from an empirical perspective. However, not all genes with high scores were selected in the 175-gene group. Because our linear program aims to reconstruct the tree with a small number of genes, it is no surprise that some potentially relevant genes are omitted.

3.4 Reconstruction is robust to noisy data

As with any learning method, it is desirable to understand how noise or uncertainty in the data can affect results. As mentioned above, the Novershtern data include at least four replicate arrays for each cell type. Although it is well-established that multiple arrays are crucial for reliably estimating differences in gene expression, we took advantage of these replicates to simulate noisy data. Ten times, we chose a single array to represent expression in each cell type, and used our approach to construct a weighted Euclidean metric. We then tested each metric by using it to construct minimum spanning trees on 100 other random combinations of replicates, for a total of 1000 reconstruction tests. The error in each tree was quantified by

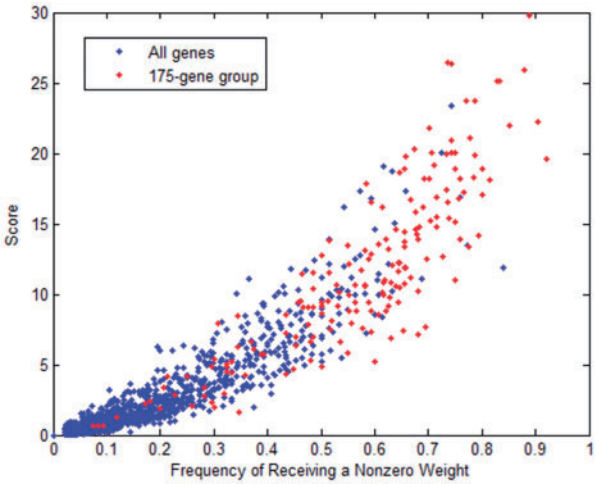

Fig. 3. Score in relation to the frequency of receiving a non-zero weight for each gene in metrics constructed from size-7000 random subsets of the genes. Frequency represents the number of solutions in which a gene receives a non-zero weight normalized by the number of times it was selected among the random 7000 genes (Color version of this figure is available at *Bioinformatics* online.)

calculating the mean number of nodes on the path from each cell type to its true parent. The average error for all 1000 trees was 2.61 (SE $\pm$ 0.03), which is less than the errors in the trees produced by all other methods we tried, including: minimum spanning trees with Euclidean distance (average error 3.96 $\pm$ 0.05), average-linkage hierarchical clustering based on Euclidean distance (average error 6.66 $\pm$ 0.04), and minimum spanning trees after projecting onto principle component (average error 10.95 $\pm$ 0.08). Thus, although we do not always get a correct tree when training and testing on noisy datasets, the trees produced by the learned metric are substantially more accurate than those produced by other approaches.

3.5 Predicting cell type locations within the hierarchy

Although the purpose of our method is to identify relevant genes that allow for reconstruction of the differentiation hierarchy, it is also interesting to evaluate its performance in a semi-supervised setting. Therefore, we tested our method 38 times, without any modifications, each time removing one cell type from the tree and learning a weighted metric from the resulting tree. We then reconstructed the tree as a MST using the learned metric on the expression data of all 38 cell types, including the one that was left out when learning the metric. When E1, E5, MG2 and T1 were excluded from the learning process, our method correctly placed them in the tree. Although it did not succeed in placing the other 34 cell types in the correct location, 26 of them were placed in the correct group/lineage of cells. Again, we quantified the error in each of the 34 trees by calculating the mean number of nodes on the path from each cell type to its true parent. The average error in all 34 trees was 0.0898, which means that, on average, the total number of nodes separating each cell type from its true parent in the tree that was constructed after including the left-out cell type is $0.0898 \times 37 = 3.32$. Therefore, with the proper modifications, our method has the potential to predict missing states or to insert new states into a given tree.

4 Conclusions and future work

We have shown that reconstruction of differentiation trees from gene expression profiles of cells at multiple differentiation stages is possible by selecting proper weights for genes to participate in a

weighted Euclidean distance metric. In applying our method to the Novershtern data, we showed that selected genes are relevant to the cell types and biological processes active throughout their proliferation and differentiation. Thus, our method offers a new way to characterize potentially biologically meaningful relationships between gene expression and differentiation trees.

Our approach works by deriving a sufficient set of constraints on the weights of the metric. Not all constraints are necessary to reconstruct the tree, although for the hematopoiesis data we found that the tree could not be reconstructed until roughly 99% of the constraints were satisfied (Supplementary Fig. S12). 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. One arbitrary choice we made in our analysis 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 linear program 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. Genes in the solution can most likely 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 which weights, if we restrict attention to known marker genes? Or to transcription factors, or signaling pathway genes, or metabolic genes, and so on?

The gene weights obtained by our method are global in the sense that the differential expression of a gene between any two pairs of cell types is weighted with a fixed constant. Our work could be extended by looking for distance metrics that are non-linear. In effect, this could allow the importance of a gene to depend on which other genes are expressed or not expressed. This is a potentially important extension because it mimics the way established cell type criteria operate (Fig. 1), with some marker genes' expression only relevant within certain lineages. The constraints we developed in Section 2.1 could still be used to guide discovery of non-linear distance metrics, but as the problem would be non-linear, an alternative to our linear programming approach for satisfying those constraints would have to be found.

Our method 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.

Funding

This work was supported by a Discovery Grant from the Natural Sciences and Engineering Research Council of Canada to T.J.P. and a Queen Elizabeth II Graduate Scholarship in Science and Technology to M.A.G.

Conflict of Interest: none declared.

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**, 889–895.

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

Ashburner,M. *et al.* (2000) Gene ontology: tool for the unification of biology. *Nat. Genet.*, **25**, 25–29.

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

Breiman,L. (2001) Random forests. *Mach. Learn.*, **45**, 5–32.

Globerson,A. and Roweis,S. (2005) Metric learning by collapsing classes. *Adv. Neural Inform. Process. Syst.*, **18**, 451–458.

Huang,D.W. *et al.* (2009) Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nat. Protoc.*, **4**, 44–57.

Joshi,A. and Berthold,G. (2011) Maximum parsimony analysis of gene expression profiles permits the reconstruction of developmental cell lineage trees. *Dev. Biol.*, **353**, 440–447.

Kanehisa,M. and Goto,S. (2000) KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.*, **28**, 27–30.

Kluger,Y. *et al.* (2004) Lineage specificity of gene expression patterns. *Proc Natl Acad Sci USA.*, **101**, 6508–6513.

Kwok,J.T. and Tsang,I.W. (2003) Learning with idealized kernels. In: *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC.* pp. 400–407.

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

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

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

Weinberger,K. *et al.* (2006) Distance metric learning for large margin nearest neighbor classification. *Adv. Neural Inform. Process. Syst.*, **18**.

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

Xing,E.P. *et al.* (2003) Distance metric learning with application to clustering with side-information. In: *Advances in Neural Information Processing Systems*, Vol. **15**, 521–528.

Yang,L. and Jin,R. (2006) *Distance Metric Learning: A Comprehensive Survey*. http://www.cs.cmu.edu/~liuy/frame_survey_v2.pdf

Ying,Y. and Li,P. (2012) Distance metric learning with eigenvalue optimization. *J. Mach. Learn.Res.*, **13**, 1–26.

Zola,H. *et al.* (2007) CD molecules2006—human cell differentiation molecules. *J. Immunol. Methods*, **319**, 1–5.