

A bunch of random studies on cancer

Adrin Jalali

October 10, 2015

Contents

1	Introduction	1
2	Background	3
2.1	Machine Learning	3
2.1.1	Empirical Risk Minimization	3
2.1.2	Feature Selection	4
2.1.3	Classification	5
2.1.4	Regression	5
2.1.5	Regularization	5
2.1.6	Support Vector Machines	6
2.1.7	Gaussian Processes	9
2.1.8	Boosting and Ensemble Methods	9
2.2	Graphs	10
2.2.1	The Shortest Path Problem	10
2.2.2	The k Shortest Path Problem	11
2.3	Lymphoma	12
3	Flow Cytometry Analysis	13
3.1	Flow Cytometry	13
3.2	Data Preprocessing and Challenges	15
3.3	High Dimensional Analysis and Visualization	15
3.3.1	flowType	15
3.3.2	RchyOptimyx	18
3.4	Lymphoma Diagnosis Quality Checking	35
4	Adaptive Learning	39
4.1	Challenges in Cancer Data	39
4.1.1	Cancer Heterogeneity	39
4.1.2	Batch Effects and Noise	39
4.2	RatBoost	39
4.2.1	Background	39
4.2.2	Methods	40
4.2.3	Results and discussion	47
4.2.4	Conclusions	52
4.3	Raccoon	52

5	Conclusion	53
A	RchyOptimyx Appendix	55

“Growth for the sake of growth is the ideology
of the cancer cell.”

- Edward Abbey

1

Introduction

Cancer has been with human species throughout our 4000 years of history. Although our understanding of cancer has changed drastically through time, but its treatment stays challenging and in many cases we cannot cure it to this date. The immense frustration of dealing with cancer not only affects the patients, but also the doctors, pathologists, and oncologists treating those patients. Siddhartha Mukherjee explains the feeling with these words [65, prologue]:

...

There were seven such cancer fellows at this hospital. On paper, we seemed like a formidable force: graduates of five medical schools and four teaching hospitals, sixty-six years of medical and scientific training, and twelve postgraduate degrees among us. But none of those years or degrees could possibly have prepared us for this training program. Medical school, internship, and residency had been physically and emotionally grueling, but the first months of the fellowship flicked away those memories as if all of that had been child's play, the kindergarten of medical training.

...

The stories of my patients consumed me, and the decisions that I made haunted me. *Was it worthwhile continuing yet another round of chemotherapy on a sixty-six-year-old pharmacist with lung cancer who had failed all other drugs? Was it better to try a tested and potent combination of drugs on a twenty-six-year-old woman with Hodgkin's disease and risk losing her fertility, or to choose a more experimental combination that might spare it? Should a Spanish-*

speaking mother of three with colon cancer be enrolled in a new clinical trial when she can barely read the formal and inscrutable language of the consent forms?

2

Background

2.1 Machine Learning

Machine learning techniques are used to extract information from data, or make some predictions about the data. This chapter briefly explains methods and techniques used in, or required to understand the proceeding chapters.

We can recognize two groups of learning problems: supervised and unsupervised learning. Supervised learning deals with data sets that are in the form of a set of input and outputs, and the task at hand is to predict the output using the input [48, TODO: chapter]. Classification and regression are supervised learning problems. Unsupervised learning, on the contrary, deals with data sets which are in the form of a set of data points, and no output is given. Clustering the most studied unsupervised learning problem [48, TODO: chapter].

2.1.1 Empirical Risk Minimization

As mentioned above, supervised learning deals with predicting an output $y_i \in Y$ given an input $x_i \in X$. The task is to find a function $f(\cdot)$ which best predicts the output, given any possible input. Because in practice we have access to a limited given data set, the best we can do is to find a function that best predicts the output given any input in the data set.

We can formulate finding the best function $f(\cdot)$, as finding a function that has the minimum loss over the data. Therefore we need a loss function which is defined as $L(x, y, f(x))$, with x the input, y the desired output, and $f(x)$ the predicted output. The loss value has to be in $[0, \infty)$, and $L(x, y, y) = 0$ [91, p. 62]. The empirical risk function is then defined as [91, p. 67]:

$$R_{emp}[f] := \frac{1}{m} \sum_{i=1}^m L(x_i, y_i, f(x_i)) \quad (2.1)$$

Now let \mathcal{F} be the function space available to choose $f(\cdot)$ from it. The best function is one which minimizes the risk function [91, p. 67]:

$$\arg \min_{f \in \mathcal{F}} R_{emp}[f] = \arg \min_{f \in \mathcal{F}} \frac{1}{m} \sum_{i=1}^m L(x_i, y_i, f(x_i)) \quad (2.2)$$

But there is a problem with the above formulation, if the function space \mathcal{F} is rich enough to fit to the given data too well. Imagine a function that returns y_i for each x_i in the training set, and 0 otherwise. This function clearly has a minimum loss of 0, but does not generalize on unseen data. This is called overfitting in machine learning. One way to fix this issue is to regularize the loss function in some way, and give preference to functions $f(\cdot)$ with lower complexity, or smoother functions. This is referred to as regularized empirical risk minimization [91, Ch. 4.1], or structural risk minimization [98, Ch. 4.1]. Assume $\Omega(f)$ is a penalty assigned to function $f(\cdot)$; then the regularized empirical risk is formulated as:

$$R_{emp}[f] := \frac{1}{m} \sum_{i=1}^m L(x_i, y_i, f(x_i)) + \lambda \Omega(f) \quad (2.3)$$

2.1.2 Feature Selection

Feature selection is the task of selecting important features to the problem at hand. It becomes particularly a hard task when the number of features in the data is of a higher magnitude compared to the number of given samples. Table 2.1 shows an example number of samples vs. number of features in a typical data. One of the challenges when dealing with such a large number of features is that if enough number of features have a probability distribution independent of the outcome, some of them might falsely seem correlated with the outcome. Another obstacle comes from the fact that our features are not independent and they function in complex networks. As a result, features should be considered in groups, which is a combinatorial and intractable problem.

Sample Data				
Sample Count	Gene Expression Data	450K Methylation	Chip	
	Feature Count	Data Feature Count		
500	$\approx 20,000$	$\approx 450,000$		

Table 2.1: An example number of samples and features in our usual data

We have used correlation [72], mutual information [87], and l_1 -regularized methods [70] as techniques to select features.

2.1.3 Classification

Classification is the problem of putting data into different classes [48]. During the training phase, the matrix $X_{samples \times features}$ is given as the input and $y_{samples}$ as the desired output. The vector y has values from a discrete set. If the set has only two distinct values, the problem is called a binary classification.

Logistic regression [103, 25], Support Vector Machines (SVM) [99, 13], and decision trees [48, TODO:chapter] are examples of classification methods.

2.1.4 Regression

In statistics, predicting a continuous output value given an input data is called regression [48, TODO:chapter]. Regression and classification differ in their desired output type. In regression the output is continuous in contrast to classification in which the output is a discrete value.

Linear regression [48, TODO:chapter], Gaussian processes [80], and kernel based regression [91, TODO:chapter] are some available methods here.

2.1.5 Regularization

Building a machine to predict the outcome with a good performance on the training set is easy if the number of features in the data is large enough compared to the number of samples, even if features are drawn from a random background probability distribution independent of the outcome. But the trained machine will perform poorly on the unseen test samples. This phenomenon is called overfitting. One way to prevent overfitting is to select potential features before training a selected model.

Another way to tackle the problem is to reduce the complexity of the models. This can be done via regularization. The l_1 -regularization is an appropriate tool when the intention is to reduce the number of features a model takes into account for prediction as well as its complexity [70].

Assume a model minimizes a loss function $E(X, Y)$, where X is the input matrix and Y is the output vector or matrix. In most regression models the loss function is defined as:

$$E(X, Y) = \|Y - X\beta\|_2 \quad (2.4)$$

The optimization algorithm finds a β that minimizes the loss function in Formula 2.4. Having enough number of features, the optimization algorithm might find a β that gives a perfect loss, i.e. 0. But in noisy environments the resulting β is probably not the real β of the underlying model producing the data. The vector β might also have some extreme values that are likely not desired. Penalizing the *size* of β as shown in Formula 2.6 will address the abovementioned concern. The size of a vector in this context is represented by its l_1 or l_2 norm as defined in Formula 2.5.

$$\|\beta\|_p := \left(\sum_{i=1}^n |\beta_i|^p \right)^{1/p} \quad (2.5)$$

$$\begin{aligned} E(X, Y) + \alpha \|\beta\|_2 &= \|Y - X\beta\|_2 + \alpha \|\beta\|_2 \\ \text{or} \\ E(X, Y) + \alpha \|\beta\|_1 &= \|Y - X\beta\|_2 + \alpha \|\beta\|_1 \end{aligned} \quad (2.6)$$

2.1.6 Support Vector Machines

Support vector machines (SVM) can be used both for regression and classification tasks [13, 90]. As a classifier, SVM finds an optimal hyperplane to separate data points in the feature space by maximizing the hyperplane's margin to the nearest point. Therefore given a test data point, its side with regard to the hyperplane determines its class. As a regressor however, SVM finds an optimal hyperplane to interpolate given data points by minimizing the hyperplane's distance from data points.

Formally speaking, given data-set \mathcal{D} of n data points:

$$\mathcal{D} = (\mathbf{x}_i, y_i) | \mathbf{x}_i \in \mathbb{R}^p, y_i \in -1, 1_{i=1}^n \quad (2.7)$$

where \mathbf{x}_i is a real vector of length p , and y_i is either 1 or -1 . A p -dimensional hyperplane, characterized by its normal vector \mathbf{w} and its intercept \mathbf{b} , is the set of points \mathbf{x} that fit in Formula 2.8.

$$\mathbf{w} \cdot \mathbf{x} - b = 0 \quad (2.8)$$

Now consider two hyperplanes on both sides of the abovementioned hyperplane as formulated below:

$$\begin{aligned} \mathbf{w} \cdot \mathbf{x} - b &= 1 \\ \mathbf{w} \cdot \mathbf{x} - b &= -1 \end{aligned} \quad (2.9)$$

The distance between each of these hyperplanes and the one in the middle is $\frac{1}{\|\mathbf{w}\|}$. Therefore the distance between the two of them is $\frac{2}{\|\mathbf{w}\|}$. For now we assume the data is linearly separable in its feature space, i.e. there exists a hyperplane that separates the data into two classes without error. Such a hyperplane satisfies the following constraint:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1 \text{ for all } 1 \leq i \leq n \quad (2.10)$$

An optimal hyperplane is one such that it maximizes the margin; hence formulated as Formula 2.11. An illustration of the optimal solution is presented in Fig. 2.1.

$$\begin{aligned}
& \arg \max_{(\mathbf{w}, b)} \frac{1}{\|\mathbf{w}\|} \\
& \text{s.t.} \\
& y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1 \text{ for all } 1 \leq i \leq n
\end{aligned} \tag{2.11}$$

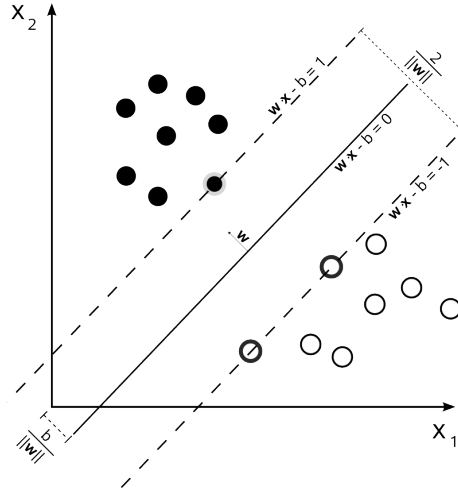


Figure 2.1: Illustration of the optimal hyperplane in a support vector machine model, for a 2-dimensional data.

However, for an easier optimization and mathematical convenience, the above optimization problem is usually formulated as Formula 2.12 which has the same solution as \mathbf{w} and b .

$$\begin{aligned}
& \arg \min_{(\mathbf{w}, b)} \frac{1}{2} \|\mathbf{w}\|^2 \\
& \text{s.t.} \\
& y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1 \text{ for all } 1 \leq i \leq n
\end{aligned} \tag{2.12}$$

which can be written as Formula 2.13 after introducing Karush-Kuhn-Tucker (KKT) multipliers [56].

$$\begin{aligned}
& \arg \min_{\mathbf{w}, b} \max_{\boldsymbol{\alpha}} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i [y_i (\mathbf{w} \cdot \mathbf{x}_i - b) - 1] \right\} \\
& \text{s.t.} \\
& \alpha_i \geq 0 \text{ for } 1 \leq i \leq n
\end{aligned} \tag{2.13}$$

Multipliers α_i will be 0 for each \mathbf{x}_i that does not lie on either of the marginal hyperplanes. For example in Figure 2.1, α_i is non-zero for only three of the data points; the ones that are exactly on either of the marginal lines. The corresponding \mathbf{x}_i for which α_i is non-zero, are *support vectors*.

It can be shown that Formula 2.14 is a dual of the optimization problem defined in Formula 2.13 [91, TODO:chapter].

$$\begin{aligned}
& \arg \max_{\boldsymbol{\alpha}} \left\{ \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \right\} \\
& \text{s.t.} \\
& \alpha_i \geq 0 \text{ for } 1 \leq i \leq n \\
& \sum_{i=1}^n \alpha_i y_i = 0
\end{aligned} \tag{2.14}$$

Now assume the following notations and definitions:

$$\begin{aligned}
\phi(\mathbf{x}) &:= \mathbf{x} \\
\langle \mathbf{x}_i, \mathbf{x}_j \rangle &:= \mathbf{x}_i^T \mathbf{x}_j \\
k(\mathbf{x}_i, \mathbf{x}_j) &:= \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle
\end{aligned} \tag{2.15}$$

Putting function k in Formula 2.14, the SVM's optimization problem can be written as:

$$\begin{aligned}
& \arg \max_{\boldsymbol{\alpha}} \left\{ \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \right\} \\
& \text{s.t.} \\
& \alpha_i \geq 0 \text{ for } 1 \leq i \leq n \\
& \sum_{i=1}^n \alpha_i y_i = 0
\end{aligned} \tag{2.16}$$

The identity function used in Formula 2.15 is not the only option. We can transform the data into another feature space using a different $\phi(\cdot)$, and then use dot-product in that space. This is useful for cases that the data is not linearly separable in its original feature space, but linearly separable using a non-linear transformation.

Using Mercer’s theorem [64] and its corollary Mercer’s condition, it can be shown that any function k satisfying the following condition can be used as a *kernel* in Formula 2.16 [91, TODO:chapter].

$$\forall \mathcal{D}, \forall c_i, c_j \in \mathbb{R} : \sum_{i=1}^n \sum_{j=1}^n k(\mathbf{x}_i, \mathbf{x}_j) c_i c_j \geq 0 \quad (2.17)$$

Arguably, other than dot-product, the most famous kernel function k satisfying the above condition is the *Gaussian kernel*, also known as the *radial basis function (RBF) kernel*:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp \left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2} \right) \\ \sigma \in \mathbb{R} \quad (2.18)$$

Many implementations use a different formulation which uses a different parametrization, using $\gamma = \frac{1}{2\sigma^2}$:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp \left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2 \right) \\ \gamma \in \mathbb{R}^+ \quad (2.19)$$

2.1.7 Gaussian Processes

2.1.8 Boosting and Ensemble Methods

For a given prediction problem the idea of boosting is to find an optimal combination of classifiers, also called “weak learners” [27]. There are many methods of finding the optimal combination of such weak learners, two of which are stochastic gradient boosting [40] and AdaBoost [39]. Stochastic gradient boosting tries to estimate the gradients of the loss function and train each individual weak learner in a way that best improves the loss function. AdaBoost tries to identify samples among given data samples that are harder to classify, and gives them more weight in the process of training individual weak learners. One way of improving AdaBoost is to take into account the confidences of predictions given by weak learners if possible and use estimated confidences in the voting process [85].

2.2 Graphs

A graph G is a set of vertices (nodes) V , and a set of edges $E = \{(v_i, v_j) | v_i, v_j \in V\}$ that connect vertices in V . A graph can be directed or undirected. In directed graphs, edges have direction, i.e. edge (s, t) is different than the edge (t, s) . In other words, the following list shows the possible sets of edges regarding vertices s and t in a directed graph:

$$\begin{aligned} E &= \{\} \\ E &= \{(s, t)\} \\ E &= \{(t, s)\} \\ E &= \{(s, t), (t, s)\} \end{aligned} \tag{2.20}$$

In an undirected graph however, edges (s, t) and (t, s) are identical, and the can in fact be represented as a set s, t instead of an ordered pair.

Graphs can also be weighted or not. If a graph G is weighted, then there is a weight assigned to each edge of the graph. We use $w_{s,t}$ to note the weight of the edge (s, t) . In undirected graphs, $w_{s,t}$ is always the same as $w_{t,s}$. Sometimes the weight of an edge is referred to as the length of an edge and noted as $l_{s,t}$ depending on the context in the literature. A graph G is connected if there is at least one path between every given two vertices on the graph.

2.2.1 The Shortest Path Problem

The *shortest path problem* is to find a path between two vertices s and t such that the total weight of the path is the minimum. In unweighted graphs, the weight of each edge is considered to be 1. Figure 2.2 highlights the shortest path between vertices A and F on the given weighted directed graph.

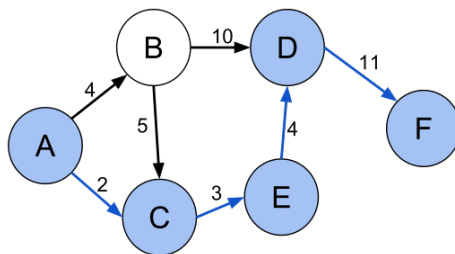


Figure 2.2: A given weighted directed graph and the highlighted shortest path between vertices A and F .

Some prominent algorithms to solve the shortest path problem are Dijkstra's [28], Bellman-Ford [8], and Floyd-Warshall [35] algorithms. Dijkstra's algorithm applies to the single-source shortest path problem on graphs with non-negative weight values with time complexity $O(|E| + |V| \log |V|)$ [38], whereas

Bellman-Ford algorithm works on graphs with also negative weights having time complexity $O(|V||E|)$. Floyd-Warshall algorithm, on the other hand, applies to all pairs shortest path problem, i.e. it finds shortest paths between all pairs of vertices, on graphs with negative and non-negative weight values and its time complexity is $O(|V|^3)$.

2.2.2 The k Shortest Path Problem

The k shortest path problem is to find not only one, but k paths from s to t such that their corresponding total weight is the least k among all distinct possible paths from s to t . Two settings of the problem are due to whether or not allowing loops in the paths. For the case when the goal is to find k best shortest paths from a single source to all other nodes, Jin Y. Yen published an algorithm of the time complexity $O(k|V|(|E| + |V|\log|V|))$ in 1971 for the loopless setting which still has the best available time complexity available [108]. It is possible to achieve better worst case time complexity if we let loops in the paths. In 1998 Eppstein came up with an algorithm with $O(|E| + |V|\log|V| + |V|k)$ time complexity, and $O(|E| + |V|\log|V| + k)$ if the problem is reduced to the single source single destination case [32]. There has been improvements to the Eppstein's algorithm, but the worst case time complexity has not been improved.

In our case, the graph is a directed acyclic graph (DAG), i.e. there are no loops in the graph. Therefore despite we require paths to be loopless, Eppstein's algorithm is sufficient and gives desirable paths. Here we give an intuitive overview of the algorithm and postpone our use case in detail to Chapter 3.

First we need to introduce some concepts and notations, and for the sake of easier reference to the original work, we keep the notation as the work done by Eppstein. Assume the problem is to find the k shortest paths from s to t on a connected directed graph G . Then consider the following:

- T : a single destination shortest path tree and t be its destination, i.e. T includes all vertices of G and a shortest path from each node to t .
- $d(v_i, v_j)$: the weight of a shortest path from v_i to v_j , or in other words the distance between the two vertices.
- $head(e), tail(e)$: if e is (v_i, v_j) , $head$ and $tail$ of e are v_i and v_j respectively.
- $l(e)$: weight or length of edge e .
- $\delta(e)$: intuitively the cost of including e in a shortest path to t , defined as:

$$\delta(e) = l(e) + d(head(e), t) - d(tail(e), t) \quad (2.21)$$

If the edge e is not a part of T , it is a *sidetrack* and the cost of including it in a path to t is non-negative [32, Lemma 1].

A key point to understanding the algorithm is the way paths are represented. A path p from s to t can be represented by the list of *sidetrack* edges it includes.

If the path p includes only one *sidetrack* edge (v_i, v_j) , it means the path is the shortest path from s to v_i , then the edge (v_i, v_j) , and then the shortest path from v_j to t . The set $sidetracks(p)$ includes all edges in p that are not in the shortest path tree T , i.e. they are in $G - T$.

To calculate the length of the path p we have [32, Lemma 2]:

$$l(p) = d(s, t) + \sum_{e \in sidetracks(p)} \delta(e) \quad (2.22)$$

Let $S = sidetracks(p)$ to be the sequence of edges of p that are in $G - T$. Then we define $path(S)$ to be the path p . We also define $prefix(S)$ to be the sequence of edges in S except the last one. Therefore $prefix(S)$ can define a path as $prefpath(p) = path(prefix(S))$.

Next we have: if the path p is from s to t in G and has a nonempty $sidetracks(p)$, then $l(p) \geq l(prefpath(p))$ [32, Lemma 3]. As a corollary of Lemma 2 and 3 we can construct a natural tree of paths which is also a heap style tree. It is a tree in a way that each node is a path p , and it has all possible paths p' for which $prefpath(p') = path(p)$. It is also a heap style tree in a way that the length of a parent node is less than or equal to all its children.

To overcome this challenge each path p , roughly speaking, is replaced by a heap of the edges that have tails on the path from $head(lastedge(p))$ to t and ordered by $\delta(e)$. Then using two intermediate directed acyclic graphs $D(G)$ [32, Lemma 4] and $P(G)$ [32, Lemma 5], a heap $H(G)$ [32, Lemma 6] is constructed with the following properties:

- $H(G)$ is a 4-heap;
- There is a bijection mapping between nodes in $H(G)$ and $s - t$ paths in G ;
- The length of an $s - t$ path in G is $d(s, t)$ plus the weight of the corresponding node in $H(G)$.

Finding k smallest nodes in a min-heap costs $k \log k$, which can be further improved by Frederickson's technique [37], and hence the time complexity of the Eppstein's algorithm [32, Lemma 7].

Although Eppstein's algorithm has the best know worst-case time complexity, it can be shown that in practice we can achieve faster running times by constructing some parts of the algorithm's intermediate structures as they're needed. In 2003 Víctor M. Jiménez and Andrés Marzal published the modified version of the algorithm and a more detailed explanation of the Eppstein's algorithm [52].

2.3 Lymphoma

3

Flow Cytometry Analysis

Here we talk about flow cytometry data, how we analyze and visualize it; and how we use that analysis alongside with some machine learning tools to classify samples into cancer subtypes.

3.1 Flow Cytometry

Flow cytometry is a technology that allows measurement of biomarkers inside and outside cells on a single cell basis [?]. The technology can also sort and separate certain cells according to a given criterion [41].

Cell preparation in flow cytometry involves suspension of the cells in a liquid containing biomarker reagents. Reagents are marked antibodies that can be detected by the laser beams in the flow cytometer machine [?]. The antibodies are usually marked with a fluorescent label. Each fluorescent marker has a corresponding peak excitation and emission wavelength which can be detected using lasers or lamps available on the flow cytometer machine. The combination of markers has to be chosen such that their corresponding wavelengths have minimal overlap; otherwise they cannot be distinguished from one another due to interference between them.

In a flow cytometer cells flow in a liquid stream one by one, where a lamps or laser beams in conjunction with sensors measure the intensity of reflected light from the cells. These measurements can be in linear or logarithmic space [89]. The measured values depend on the light intensity projected onto cells which can be tuned by changing the voltage of the lasers or lamps. Different wavelengths correspond to different markers, but they might overlap. When the tail of the emission spectrum of a marker overlaps with the main part of the emission

spectrum of another marker, it is called spillover as shown in Fig. 3.1 [82].

Compensating for spillover requires a spillover matrix (SM). $SP_{i,j}$ shows the percentage that marker i spills over marker j . The compensation matrix (CM) is then calculated as the inverse of the spill over matrix. Let S be the true signal value, and O be the observed value. Then we have ¹:

$$\begin{aligned} CM &= SM^{-1} \\ S &= O \times CM \end{aligned} \tag{3.1}$$

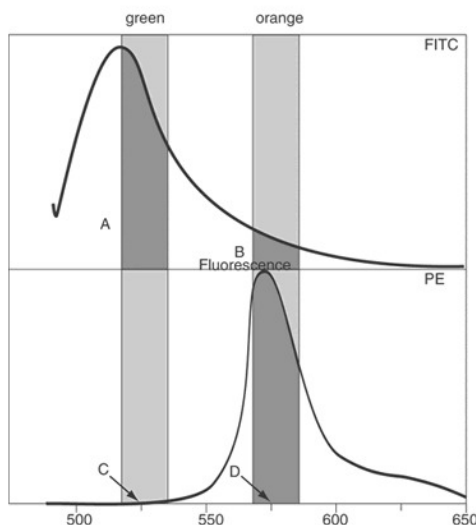


Figure 3.1: [DIRECT QUOTE] Fluorescence emission spectra for FITC and PE. The emission spectrum (the wavelengths of light generated by excitation of these molecules) is shown for an excitation at $488nm$ (the same as the argon-ion laser line). FITC emission is maximal at $\sim 515nm$; typically, a filter centered on $530nm$ is used to collect the emitted light (shaded region). The emission of is farther red, with a maximum at $\sim 575nm$; typically, a filter centered on this emission maximum is used to collect. Note that PE has some emission in the wavelength bands used to collect PE fluorescence (B); typically, the amount of light in the $575nm$ band is $\sim 15\%$ of that in the $530nm$ band (A). The PE has very little emission in the $530nm$ band (C), usually less than 2% of the emission in the $575nm$ band (D) [82].

¹<http://bioinformin.net/cytometry/compensation.php>

3.2 Data Preprocessing and Challenges

Transformation and spillover compensation are the two main phases of raw flow cytometry data preprocessing.

Transformation: The measured fluorescent intensities almost exponentially correspond to the number of existing fluorescent markers on or inside the cell. Therefore a proper transformation of the raw data is essential in order to have the data in a linear space. Logarithmic, log-linear hybrid transformation Logicle [71], and hyperbolic arcsine [?] are some commonly used transformations. Some studies have compared different transformation techniques and reported their advantages and disadvantages [34, 77].

Spillover Compensation: Compensation is done as shown in Formula 3.1 and it relies on a given compensation or spillover matrix.

In practice data are produced through time and also maybe in different labs. This means reagent batches are different, and also flow cytometry machines are not necessarily calibrated alike, which also affects compensation matrices. Therefore normalization is a crucial step to make samples comparable [47].

3.3 High Dimensional Analysis and Visualization

Manual analysis of flow cytometry data involves *gating*. Researchers use density or scatter plots of one or two selected dimensions of flow cytometry data in order to visualize and also select some areas on those plots to further investigate cells within the selected area. Visualization and further gating of those selected cells is commonly a next step to the analysis.

Manual gating of cells across several samples is a labor intensive and time consuming process. Not being able to analyze the data in its original higher dimensional space is another disadvantage of manual flow cytometry data analysis.

3.3.1 flowType

Flow cytometry has undergone a “chromatic explosion” over the past decade and can now measure 17 markers at once for each of hundreds of thousands of individual cells [22]. Since then, mass cytometry has enabled measurement of 30–45 markers per cell [11], while single-cell multiplexed RT-qPCR can measure 50–96 mRNAs per cell [106]. The growth in high-throughput single-cell data continues to outpace development of corresponding bioinformatics techniques [22]. To answer this challenge, we previously developed flowType [2] and RchyOptimyx [4]. flowType uses partitioning of cells, either manually or by clustering, into positive or negative for each marker to enumerate all cell types in a sample, e.g. [3]. RchyOptimyx measures the importance of these cell types by correlating their abundance to external outcomes, such as disease state or patient survival, and distills the identified phenotypes to their simplest possible form. These

packages have been used to identify several novel cell populations correlated with HIV outcome [2]. More recently, this pipeline has been used to evaluate standardised immunological panels [102], to optimise lymphoma diagnosis [26], and to analyse a range of other clinical data (unpublished).

However, the higher dimensionality of data produced by mass cytometry generates up to $3^{45} \approx 10^{21}$ possible cell types, with an even greater number (up to $3^{96} \approx 10^{45}$) for single-cell qPCR; these magnitudes are beyond the capabilities of flowType and RchyOptimyx. Furthermore, flowType and RchyOptimyx have thus far only treated cells as being either positive or negative for a marker. In practice, many biomarkers can have a range of expression levels such as “dim” and “bright”. In this application note, we detail architectural improvements to flowType and RchyOptimyx to overcome these limitations.

Approach

Our primary challenge was to enable flowType to generate a number of cell types tractable on most common workstations (e.g. those with 4–12GB of RAM). We hereafter denote the original flowType implementation as flowType-BF (brute force), and the new version as flowType-DP (dynamic programming). Whereas flowType-BF completely enumerates all cell types over all $[1, \dots, m]$ markers, we opted in flowType-DP to use a breadth-first strategy of enumerating all cell types defined over a subset of $k \leq m$ markers. We provide a memory use estimation function, to assist users in finding a k that fits within the limits of their hardware. To improve computation time, in flowType-DP we implemented a dynamic programming approach, which exploits the fact that cell types can be arranged into a hierarchy, and membership of any given cell type over n markers is equal to the intersection of one of its parent types (over $n - 1$ markers) with a single-marker cell type. flowType-DP first enumerates all cell types involving only 1 marker by simple partitioning and then iterates over $2, \dots, k$ markers, computing all cell types for each level n by set intersections between corresponding cell types in levels $n - 1$ and 1.

For example, membership of the cell type $\text{CD45}^{++}\text{CD117}^{+}\text{CD34}^{-}$ is computed as follows:

$$\begin{aligned} & \{\text{CD45}^{++}\text{CD117}^{+}\text{CD34}^{-}\} \\ &= \{\text{CD45}^{++}\text{CD117}^{+}\} \cap \{\text{CD34}^{-}\} \\ &= \{\text{CD45}^{++}\} \cap \{\text{CD117}^{+}\} \cap \{\text{CD34}^{-}\} \end{aligned} \tag{3.2}$$

To allow partitioning into levels other than positive and negative, we used a string representation for cell types. The string has one integer character for every marker, denoting the partition, or zero if the marker is not used. Values $1, \dots, n$ denote partitions 1 to n . For example, if the set of markers were $\{\text{CD3}, \text{CD45}, \text{CD13}, \text{CD117}, \text{CD34}\}$ the cell type $\text{CD45}^{++}\text{CD117}^{+}\text{CD34}^{-}$ would be represented by 03021. RchyOptimyx uses a dynamic programming algorithm for efficiently constructing k -shortest paths [32]. We modified RchyOptimyx’

graph construction component to be able to handle more than one partition per marker.

Results and Discussion

We evaluated flowType-DP against flowType-BF on a 10-marker dataset available from Flow Repository (ID FR-FCM-ZZZK) [2]. flowType-DP showed a substantial speedup over flowType-BF, which increases exponentially with the number of cells and markers. For example, at 10^6 cells and 10 markers, flowType-DP is 14 times faster (see Fig. 3.2a and b). Comparison on larger datasets was not possible, due to the limitations of flowType-BF.

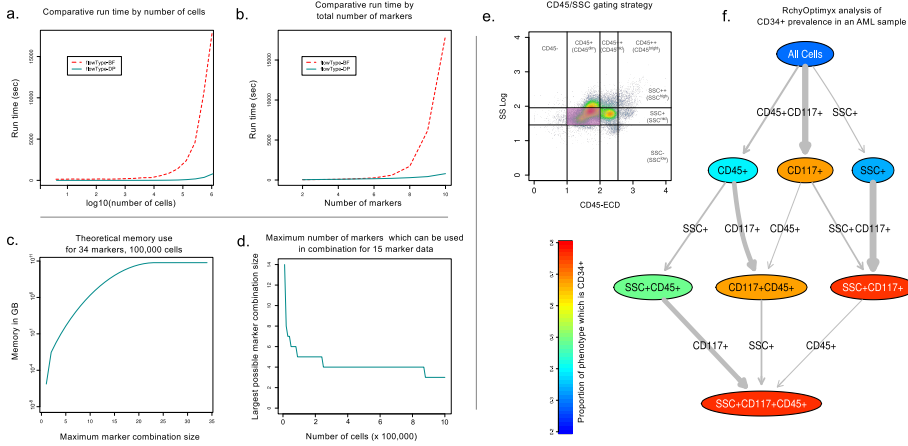


Figure 3.2: **a-b. Run time comparison of flowType-DP to flowType-BF** in terms of number of cells (a) and number of markers (b). **c-d. Possible thresholds for marker combinations using flowType-DP** for typical mass cytometry data (c) and polychromatic flow cytometry data (d). **e-f. Three/four-partition flowType-generated, RchyOptimyx-visualized cell type hierarchy** on a bone marrow sample from a patient with AML. Cell population identification strategy used for SSC and CD45, with the CD34-enriched subset highlighted (e). RchyOptimyx analysis showing CD34 enrichment (f).

We also computed the limits for k on a hypothetical machine with 12GB of RAM for samples representative of mass cytometry (Fig. 3.2c) and polychromatic flow cytometry (Fig. 3.2d), both of which would be intractable for flowType-BF. flowType and RchyOptimyx are now able, within the memory of a common workstation (12GB), to analyze 34-marker data.

Finally, to demonstrate the importance of several partitions per marker, we applied flowType and RchyOptimyx to an acute myeloid leukemia sample from Flow Repository (ID FR-FCM-ZZYA) (Fig. 3.2e-f). CD34 is a stem-cell marker typically expressed on AML blast cells. These blasts are also known

to have dimly positive CD45 expression and low SSC [101]. By partitioning CD45 and SSC into four and three partitions, and naively running flowType and RchyOptimyx to search for CD34-enriched cell types, we were able to find that the $\text{SSC}^{\text{low}}\text{CD45}^{\text{dim}}$ cell type had a high proportion of CD34^+ cells, as expected. This would not have been possible with only two partitions for each of CD45 and SSC.

3.3.2 RchyOptimyx

Recent advances in FCM instrumentation and reagents have enabled high-dimensional analyses to identify large numbers of cell populations with potentially significant correlations to an external outcome. However, studies often fail to characterize the complex relationships between the markers involved in the identification of these cell populations. Revealing this information can provide additional insight into the biological characteristics of the populations identified. The choice of markers for new panels has been a source of ongoing debate, including efforts such as the Human ImmunoPhenotyping Consortium (HIPC), the Federation of Clinical Immunology Societies Federation of Clinical Immunology Societies (FOCiS) sponsored Flow Immunophenotyping Technical Meetings (FITMaN), and the Optimized Multicolor Immunophenotyping Panels (OMIPs) articles [61, 84, 62, 21, 104, 12, 36, 66, 31, 112, 57, 75]. Understanding the relationships between the markers involved in identification of the target cell population and the characteristics of that cell population (*e.g.*, its correlation with a clinical outcome) is fundamental to the design of effective marker panels. For example, one could use a high-dimensional flow or mass cytometry assay to measure a large list of candidate markers. However, this can result in parsing the cells into (*e.g.*, clinically) redundant subsets [9]. Excluding these redundancies (*e.g.*, markers less important for prediction of a clinical outcome) will result in a panel of the most clinically relevant markers.

High dimensional FCM data is usually analyzed using a laborious sequential manual analysis procedure in which a series of thresholds or 2-dimensional polygons (or gates) are applied to histograms or scatter plots of markers (*e.g.*, [73, 43]). However, manual gates provide little insight into the relative importance of each gate to the final results. For example, consider a six color assay with markers named 1 to 6. If the expression of each marker is considered to be on, off, or does not matter (*e.g.*, markers named 1, 2, and 3 in phenotype 1^+2^- , respectively), a total of $3^6 = 729$ cell populations can be distinguished based on these markers. A given immunophenotype involving all six of these markers (*e.g.*, $1^+2^-3^+4^-5^+6^-$) can have $2^6 = 64$ parent populations (*e.g.*, 1^+ , 1^+2^-). Quantifying the relationship between the cell population of interest and these parent populations is fundamental to our understanding of the importance of the markers for different gating strategies. The order in which the gates are applied to the data is not important, as long as all of the gates are used (*i.e.*, sequential gating is commutative). However, to decrease the size of the marker panel, the relative importance of the gates should be determined. For example, the measurement of the phenotype mentioned above using only five colors requires

the determination of the importance of each marker to identify and remove the least important one (*i.e.*, the identification of the parent population with five markers that is most similar to the original phenotype). This is further complicated by the fact that some cell populations can be identified using more than one combination of markers and gating strategy; therefore, each marker can be used in different positions in the gating hierarchy and can have different priorities, depending on the choice of the gating strategy. For example, the 3^+ gate is involved in both $1^+2^-3^+$ and $3^+4^-5^+$, both parents of the $1^+2^-3^+4^-5^+6^-$ phenotype described above. However, depending on the amount of redundancy between marker 3 and others, this marker can have different levels of importance for these two parent populations.

Another use-case for measuring the importance of the markers is the investigation of a large number of closely related phenotypes (*e.g.*, those identified by bioinformatics pipelines) by identifying their common parent populations. Several computational tools have been developed for automated identification of cell populations (*e.g.*, [60, 33, 76, 17, 68, 110, 78, 92, 5, 10, 79]) and recent studies have used these tools to identify novel cell populations that correlate with clinical outcomes (*e.g.*, [2, 109, 24, 83, 7]). In addition, the results of the FlowCAP-II project² have shown that several algorithms can accurately and reproducibly identify cell populations correlated with external outcomes. However, these algorithms provide limited information regarding the importance of the markers involved in defining the cell populations [2, 18]. This situation is even more complicated than sequential manual gating, since most of these bioinformatics pipelines work based on multivariate classifiers, and as a result, more than one cell population can be responsible for the final predictions. Therefore, markers can have different relative importance in defining the multiple cell populations within the multivariate model. Quantifying the markers for each phenotype involved in the multivariate model can provide additional insight into the differences between closely related cell populations. For example, if two phenotypes $1^+2^-3^+4^-5^+$ and $1^+2^-3^+4^-6^+$ are identified as correlates of a disease, and if markers 5 and 6 (which are the only differences between them) are the least important markers for the former and latter phenotypes respectively, then these two phenotypes are likely to correspond to the same cell population (as far as the correlation with the disease is concerned). However, if markers 5 and 6 are the most important for the phenotypes, these can correspond to two biologically different cell populations.

To address these problems, we developed RchyOptimyx, a computational tool that uses dynamic programming and optimization techniques from graph theory to construct a cellular hierarchy, providing the best gating strategies to identify target populations to a desired level of purity or correlation with a clinical outcome, using the simplest possible combination of markers.

²<http://flowcap.flowsite.org/summit2011.html>

Materials and Methods

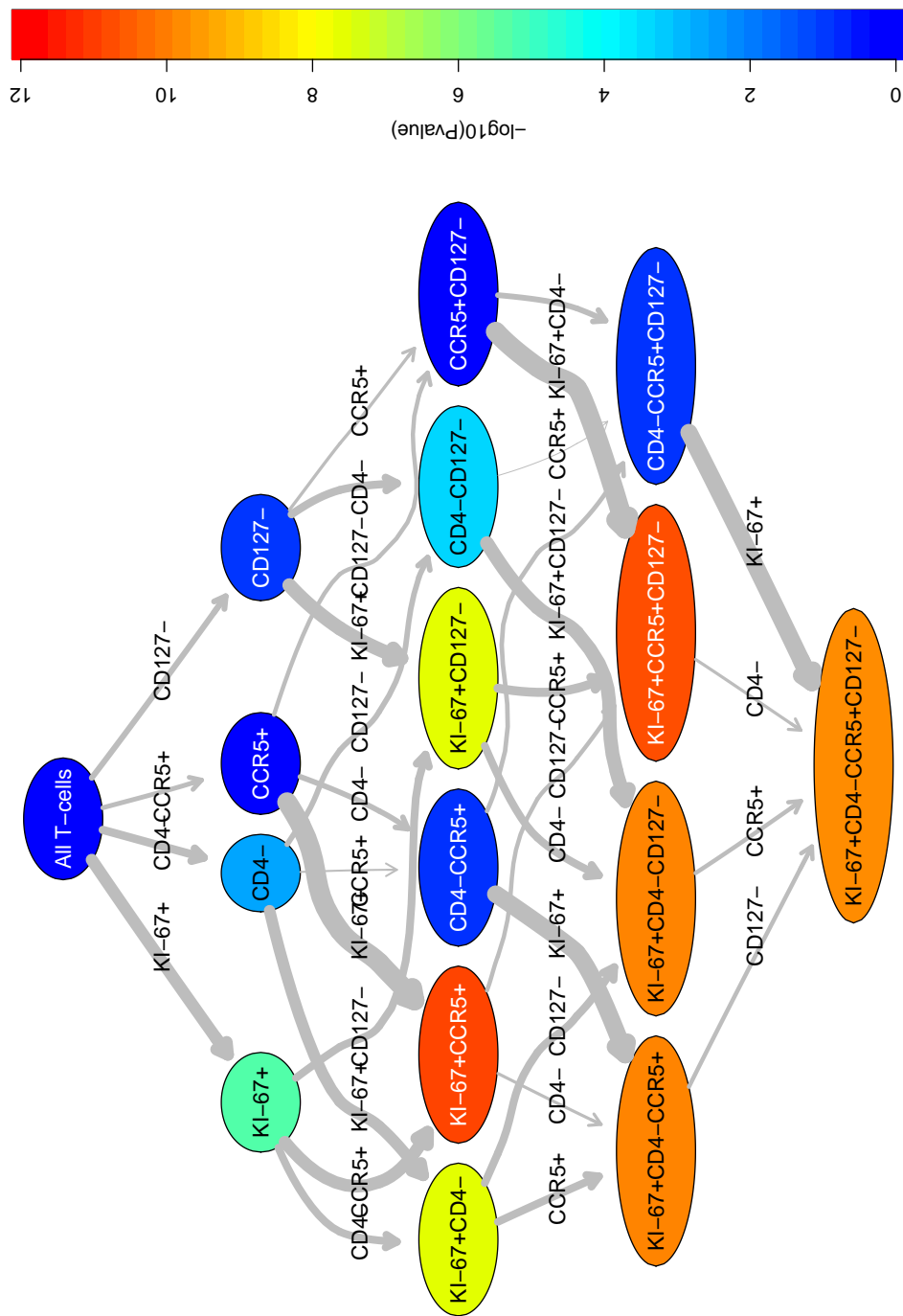
Our methodology builds on the flowType pipeline[2]. flowType comprehensively identifies cell populations defined by all possible gating strategies (hierarchies) in the data set using a partitioning strategy (*e.g.*, clustering algorithm like flowMeans [2]) and scores them by a statistical test (*e.g.*, the log rank test for difference in survival distributions). Given the list of all cell populations and their scores, RchyOptimyx uses a dynamic programming approach to find the best cellular hierarchy within a reasonable time for interactive data analysis (*e.g.* less than 2 minutes for 30 color data), as well as a number of best suboptimal hierarchies, to enable mining of the space of best gating strategies and purities for a given target cell population.

Terms and Definitions

Let \mathcal{M} be the set of m markers of interest (*e.g.*, $\mathcal{M} = \{KI-67, CD28, CD45RO\}$), a single marker phenotype be a phenotype having only one marker (*e.g.*, $CD28^+$), a phenotype P be a set of single marker phenotypes (*e.g.*, $P = KI-67^+CD28^-$), and M (not to be mistaken with \mathcal{M}) be a phenotype of size m that involves all of the markers (*e.g.* $M = KI-67^+CD28^-CD45RO^-$). The power set of M , $\mathcal{P}(M)$, is of size 2^m and contains every possible subset of M . The scoring function $S(\cdot)$ assigns a score to each member of $\mathcal{P}(M)$, such that higher values are assigned to more important phenotypes (*e.g.*, those with a stronger correlation with a clinical outcome).

Given an arbitrary M , the directed acyclic graph (DAG) G_M has $m+1$ levels from 0 to m , each level i including every member of $\mathcal{P}(M)$ of size i . Node s is connected to node t with a directed edge (s, t) if and only if $|t| = |s| + 1$ and the two associated sets of s and t differ only in one single phenotype marker (*i.e.*, t is an immediate parent of s). Let the weight of edge (s, t) be $-S(t)$ (so that paths with maximum score can be found by searching for paths with minimum total weight).

The node with 0 markers is the root (or source) node, and the node with the complete set of markers is the sink node. A path from source to sink is called a hierarchy path, or simply a hierarchy. An example of graph G_M for $M = KI-67^+CD4^-CCR5^+CD127^-$ is illustrated in Supplementary Figure 3.3.



The graph G_M has $|\mathcal{P}(M)| = 2^m$ nodes, one node for each parent phenotype of the phenotype of interest. The number of edges is equal to the number of markers (m), times the number of edges that have the specified marker. Each marker appears in 2^{m-1} nodes, therefore the number of edges is $m \times 2^{m-1}$.

A scoring function is needed to find the best hierarchy. This function should give a higher rank to hierarchies that go through more important parent populations earlier (*i.e.*, those that achieve a higher clinical significance with fewer markers). Because each node of the hierarchy is a phenotype, and each phenotype has a given score value $S(\cdot)$, we use the *total score* function $T(\cdot)$ - the sum of all negated phenotype scores in the hierarchy - as the scoring function:

$$\begin{aligned} T(\mathcal{H}) &= \sum_{(s,t) \in E_{\mathcal{H}}} W(s,t) \\ &= \sum_{(s,t) \in E_{\mathcal{H}}} -S(t) \\ &= \sum_{t \in V_{\mathcal{H}} \setminus v_0} -S(t) \end{aligned} \tag{3.3}$$

where \mathcal{H} is the given hierarchy, $E_{\mathcal{H}}$ is the set of edges of hierarchy \mathcal{H} , $V_{\mathcal{H}}$ is the set of vertices of same hierarchy, and v_0 is the first node in the hierarchy. Applying this function to G_M , the best hierarchy is the minimum weighted path in G_M . We note that, in principle, more complex functions can be used to compute the total score of a given hierarchy; for example, in applications in which phenotypes with fewer markers are more important than the other phenotypes, an exponential function can be used to increase the weight of the earlier phenotypes in the hierarchy.

Dynamic Programming to Identify the Best Hierarchy

For cell populations characterized by m markers, finding the best hierarchy by searching through all possible hierarchies would require time $O(m!)$, which is impractical for even moderately large m . To make this problem tractable using dynamic programming, we define *best total score* function $T^*(\cdot)$, which computes the score of the best hierarchy leading to the given phenotype. $T^*(\cdot)$ is defined recursively as follows:

$$T^*(P^k) = \begin{cases} -S(P^k) & \text{if } k = 1 \\ \min\{T^*(P^k \setminus P_i^k) - S(P^k) | i = 1, \dots, k\} & \text{otherwise} \end{cases} \quad , \tag{3.4}$$

where P^k is a cell population defined by k single marker phenotypes, and $P^k \setminus P_i^k$ is P^k with the i^{th} single marker phenotype removed. For example, if $P^3 = KI-67^+CD28^-CD45RO^+$, then $P^3 \setminus P_1^3 = CD28^-CD45RO^+$. In other words, there is an edge from $P^k \setminus P_i^k$ to P^k in G_M where, P^k is a subset of M . Also note that $-S(P^k)$ is the weight of the edge $(P^k, P^k \setminus P_i^k)$ in G_M .

Using dynamic programming, we calculate the value of $T^*(\cdot)$, iterating from level 0 to m on G_M . Calculating each node's score requires a number of constant

time operations equal to the number of edges entering the node. Therefore, the total number of operations is proportional to total number of edges ($m \times 2^{m-1}$), and the overall time complexity of our programming procedure for determining $T^*(.)$ values for all phenotypes in the graph is $O(m \times 2^{m-1})$. An illustration of the dynamic programming space for three dimensional space, *i.e.* having three markers, as well as two paths in that space is shown in Figure 3.4.

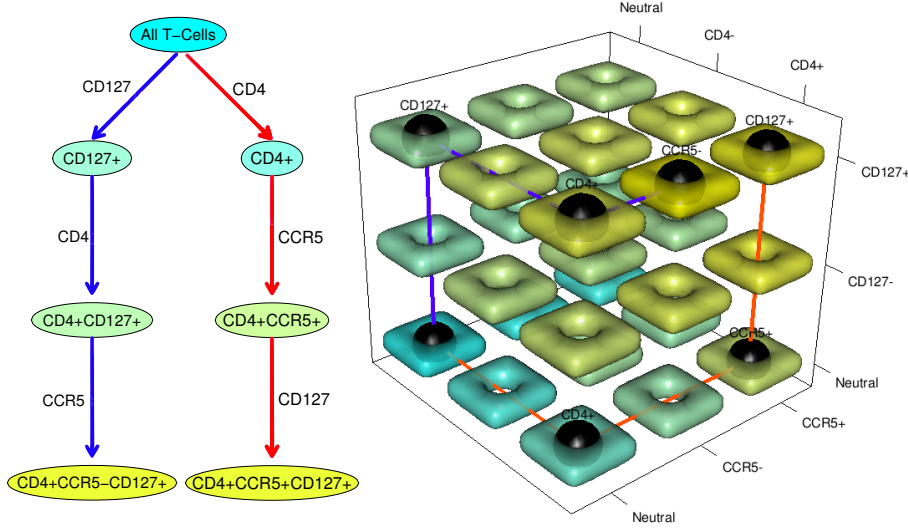


Figure 3.4: Dynamic programming algorithm for two cell populations defined by 3 markers. The best path for each of the cell population is shown in red and blue respectively. As an example, the red path ends at $CD4^+CCR5^+CD127^+$. Three markers are available to be added. First, CD4 is added (changes from does not matter to positive). Then two options will be available for the next step (CD127 and CCR5). After selection of CCR5, only one option will be left for the final step (CD127). Therefore for three markers, $\frac{3 \cdot (3-1)}{2} = 6$ comparisons were required. **Left:** A hierarchy for the two paths. The label of an edge is the name of the single marker phenotype that is the difference between its head set (s) and its tail set (t). **Right:** the dynamic programming space for the 3 markers. Black spheres mark the nodes in the dynamic programming space used by the two paths. The colors of the nodes on the left match that of the square tori on the right and correspond to the relative score of each cell population.

Search for Near-Optimal Hierarchies

The hierarchy selected by the dynamic programming algorithm is the best gating strategy for a given cell population. However, we would also like to identify alternate gating strategies with slightly less desirable scores. To find these

near-optimal paths, we reformulate the problem as identification of a desired number of minimum weight paths: In G_M , the minimum weight path from source to sink is the best hierarchy (identical to the one generated by dynamic programming). To generate additional, sub-optimal hierarchies, a list of the next minimum weight paths must also be generated. These paths can be identified using the method by Eppstein [32]. As noted in the original article, elaborating the details of this algorithm is complicated and requires substantial background in algorithm design, which is well beyond the scope of this work. Briefly, this method uses the minimum spanning tree of G_M and computes a heap structure for each node; it then merges the heaps in an efficient way to construct a 4-heap data structure. Using this 4-heap and a given arbitrary number l (the number of desired paths), it generates l -minimum weight paths in time $O(e + v + l)$ for a DAG with e edges and v nodes (see Theorem 4 of [32] for details).

Hence, the time complexity of our algorithm can be calculated based on the number of edges and nodes using the time complexity of the l -minimum weight paths method:

$$\begin{aligned} O(e + v + l) &= O(m \times 2^{m-1} + 2^m + l) \\ &= O(m \times 2^{m-1} + 2 \times 2^{m-1} + l) \\ &= O((m + 2) \times 2^{m-1} + l). \end{aligned} \tag{3.5}$$

For example, the number of operations with our approach on a dataset with $m = 10$ markers would be $\approx 10^4$ compared to $\approx 3 \times 10^6$ for the exhaustive search approach. Our method therefore takes ≈ 0.23 CPU seconds vs ≈ 69 CPU seconds for exhaustive search, run under 64 bit Linux (version 3.3) on 2.93GHz Intel Xeon CPU with sufficient memory (proportional to 2^M). For a phenotype involving $m = 20$ markers, these numbers increase to ≈ 1.2 CPU seconds vs $\approx 10^{11}$ CPU seconds (more than 4000 years), respectively. Even for a phenotype involving $m = 30$ markers measured by a CyTOF assay (mass spectrometry-flow cytometry hybrid device [69, 10, 20]), RchyOptimyx remains feasible, with a runtime of ≈ 102 CPU seconds, while the brute-force method would take $\approx 10^{22}$ CPU seconds. The final output of RchyOptimyx is the corresponding subgraph of G_M that includes all calculated paths (*i.e.*, the optimized hierarchy, *e.g.*, Fig. 3.5).

Datasets

We validated RchyOptimyx on two high-dimensional datasets, produced by mass and polychromatic flow cytometry.

Mass cytometry analysis of bone marrow cells from normal donors In this dataset, 31 parameters were measured for mononuclear cells from a healthy human bone marrow (see [10] for details). We used the results of three assays on samples subject to *ex vivo* stimulation by IL7 (measured by pSTAT5), BCR (measured by pBLNK), and LPS (measured by p-p38) as well as an unstimulated control. 13 surface markers were included in the analysis: CD3, CD45,

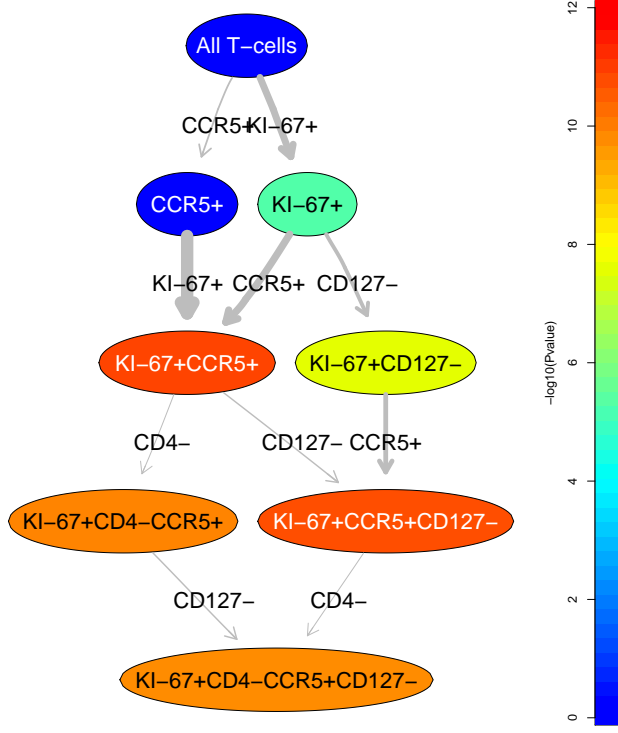


Figure 3.5: An optimized cellular hierarchy for prediction of HIV's clinical outcome using $KI67^+CD4^-CCR5^+CD127^-$ T-cells. The color of the nodes shows the significance of the correlation with clinical outcome (p-value of the logrank test for the Cox proportional hazards model) and the width of each edge (arrow) shows the amount of change in this variable between the respective nodes.

CD45RA, CD19, CD11b, CD4, CD8, CD20, CD34, CD33, CD123, CD38, and CD90. Singlets were gated manually, as described in the original publication.

Polychromatic flow cytometry analysis of HIV⁺ patients This dataset consists of 13 color PFC assays of 466 HIV⁺ subjects enrolled in the Infectious Disease Clinical Research Program's HIV Natural History Study. Basic demographic characteristics of this dataset are described elsewhere [105]. Cryopreserved peripheral blood mononuclear cells stored within 18 months of the date of seroconversion were analyzed using PFC as described by Ganesan *et al.* [42]. The cohort included 135 death/AIDS events, as defined by 1993 guidelines [16]. The date of the last follow-up or initiation of highly active anti-retroviral therapy (HAART) was considered a censoring event. CD14 and V-amine dye were used to exclude monocytes and dead cells, respectively, CD3 was used to gate T-cells. Using the staining panel and flowType, we enumerated various subsets

of naive and memory T-cells, defined by CD4, CD8, CD45RO, CD27, CD28, CD57, CCR5, CCR7, CD127, and KI-67. Using a log rank test with Bonferoni's multiple test correction, we scored each subset (cell population) in terms of its correlation with HIV progression [2].

Results

Designing a Panel to Detect a Population Expressing an Intracellular Marker using Surface Markers

In this use-case, our goal was to identify cell populations that are affected by different stimulations in the mass cytometry dataset. We used flowType to identify a list of populations that had a high overlap with either the IL3⁺, BCR⁺, or LPS⁺ populations (determined manually - see Fig. 3.6). For each cell population, this value was calculated as the difference in its intersection with the IL3⁺, BCR⁺, or LPS⁺ compartments between the stimulated and unstimulated sample. For example, for a given cell population CP, the overlap with IL3⁺ was defined as:

$$Overlap^{IL3^+}(CP) = \left(\frac{\# \text{ IL3}^+ \text{ cells in CP}}{\# \text{ cells in CP}} \right)_{stim} - \left(\frac{\# \text{ IL3}^+ \text{ cells in CP}}{\# \text{ cells in CP}} \right)_{unstim} \quad (3.6)$$

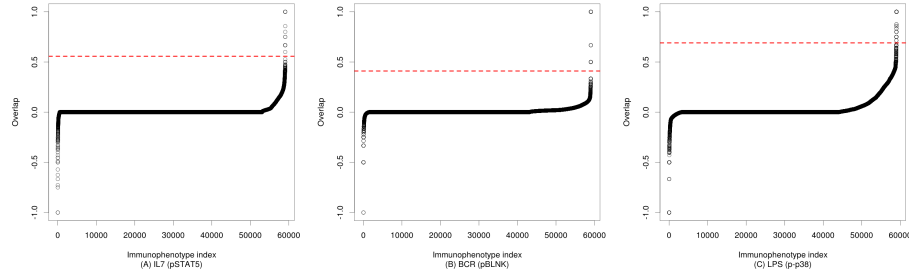


Figure 3.6: All immunophenotypes ordered by their overlap with the cell population of interest. The red dashed lines indicate the cutoffs used for selecting the immunophenotypes with “high overlap”.

The immunophenotypes with a high overlap, as identified by flowType, are listed in Tables A.1, A.2, and A.3. These immunophenotypes were analyzed using RchyOptimyx (*e.g.*, Fig. 3.7 for BCR) and then merged into a single graph, shown in Fig. 3.7. This graph suggests that T-cells (CD3⁺) followed by cytotoxic T-cells (CD3⁺CD4⁺) are the main parent populations that are affected by IL7 stimulation (panel A). As expected, BCR stimulation affected B-cells (CD19⁺CD20⁺CD3⁻), and LPS stimulation increased the proportion of CD19⁻CD33⁺CD3⁻ cells (Panels B and C, respectively). These results are

generally consistent with those reported in the original study (Figure 2 and panel C of Figure 3 of [10]).

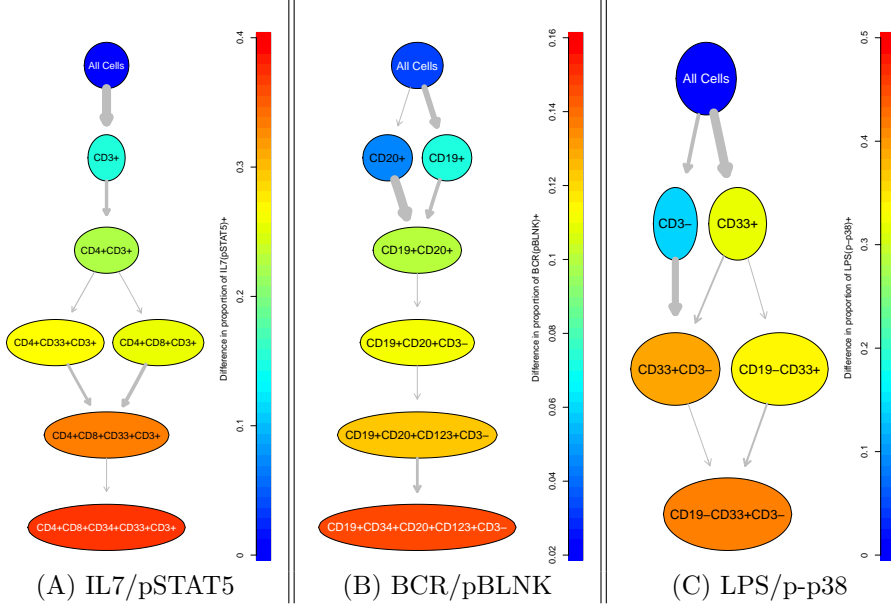


Figure 3.7: Three optimized hierarchies for identification of cell populations with maximum response to IL7, BCR, and LPS measured by pSTAT5, pBLNK, and p-p38, respectively. The colour of the nodes and the thickness of the edges shows the proportion and change in proportion of cells expressing the intracellular marker of interest, respectively.

Simplifying Gating Strategies

Here we use RchyOptimyx to demonstrate an example of the use case of establishing a simpler combination of markers that can be used to identify a target population at a desired level of purity. For analysis of the PFC dataset, Ganesan *et al.* used a strict, but potentially redundant definition for naive T-cells, of $CD28^+CD45RO^-CD57^-CCR5^-CD27^+CCR7^+$, within the $CD3^+CD14^-$ compartment [42]. The purity of a given parent cell population (CP) of this target was defined as its mean purity for the strictly-defined naive T-cells:

$$Purity(CP) = \frac{\sum_{All\ Samples} \frac{\# CD28^+CD45RO^-CD57^-CCR5^-CD27^+CCR7^+ cells}{\# cells\ in\ CP}}{\# Samples} \quad (3.7)$$

Figure 3.8 shows the results of analysis with RchyOptimyx where a combination of only three markers ($CD45RO^-CCR5^-CCR7^+$) identified the strict naive T cell population to 95% purity (within the $CD3^+CD14^-$ compartment). The

range of available purities, and determination of an appropriate cutoff is experiment dependent (*e.g.*, on the range of available markers or biological question being researched) and this result is only provided as an example of the utility.

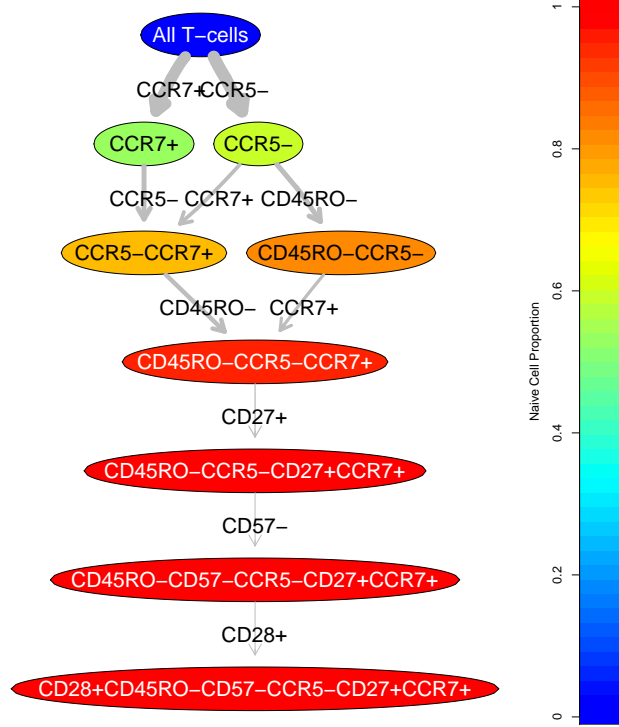


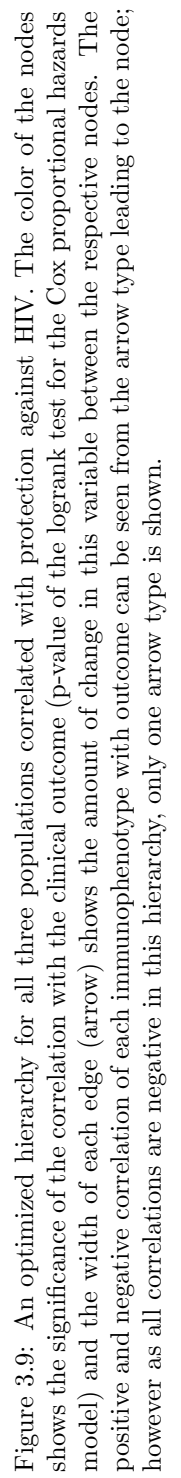
Figure 3.8: An optimized cellular hierarchy for identifying naive T-cells. The color of the nodes and the thickness of the edges shows the purity and change in purity of the original naive phenotype within the given cell population, respectively.

Characterization of a Large Number of Immunophenotypes

Here we use RchyOptimyx to demonstrate an example of the use-case of summarizing a large list of immunophenotypes of interest (as identified by a bioinformatics pipeline) into a single hierarchy using their most important common parent populations.

In a previous study of the PFC dataset, we identified 101 immunophenotypes (Table A.4) in HIV⁺ patients that had a statistically significant correlation with HIV's progression [2]. The score of each population was calculated as $-\log_{10}(p)$ where p was the p-value of the logrank test before adjustment for multiple testing (higher values represent a stronger correlation with the clinical

outcome). The 101 immunophenotypes were analyzed using RchyOptimyx and the resulting hierarchies were merged into a single graph (Figure 3.9). This graph indicated three groups of immunophenotypes that were significantly correlated with HIV's outcome (left, center, and right branches). The left branch consisted of $KI-67^+CD4^-CCR5^+CD127^-$ T-cells. These cells were thought to be statistically significant mainly because they are long-lived ($CD127^-$) T-cells with high proliferation ($KI-67^+$). RchyOptimyx showed that the significance of this population is related to the $KI-67^+CCR5^+$ compartment and not $CD127^-$ (Figure 3.9, the left branch) as the $CD127$ marker is not needed to achieve the approximately the same score. This is in agreement with the results of two recent studies [46, 51]. The terminal node of the center branch consisted of seven markers ($CD45RO^-CD8^+CD57^+CCR5^-CD27^+CCR7^-CD127^-$). RchyOptimyx revealed that its most important parent population is $CD8^+CCR7^-CD127^-$, with a weaker correlation with the clinical outcome. Finally, the right branch ($CD28^-CD45RO^+CD4^-CD57^-CD27^-CD127^-$) suggests several parent populations with minimal overlap and strong correlation with the clinical outcome (*e.g.*, $CD28^-CD4^-CD57^-CD127^-$ and $CD45RO^+CD4^-CD127^-$).



Discussion

Sequential analysis of the markers involved in manual or automated identification of cell populations is fundamental to our understanding of the characteristics of the cell population. In sequential gating, the order in which the gates have been applied does not affect the final results. However, ordering the gates by their relative importance has two use-cases: 1) identifying a cell population of interest, using the smallest possible panel of markers; 2) summarizing a long list of closely related (and perhaps overlapping) immunophenotypes by identifying their most important common parent populations. However, increasing the number of markers quickly renders this approach infeasible (*e.g.*, Fig. 3.10 for only six markers).

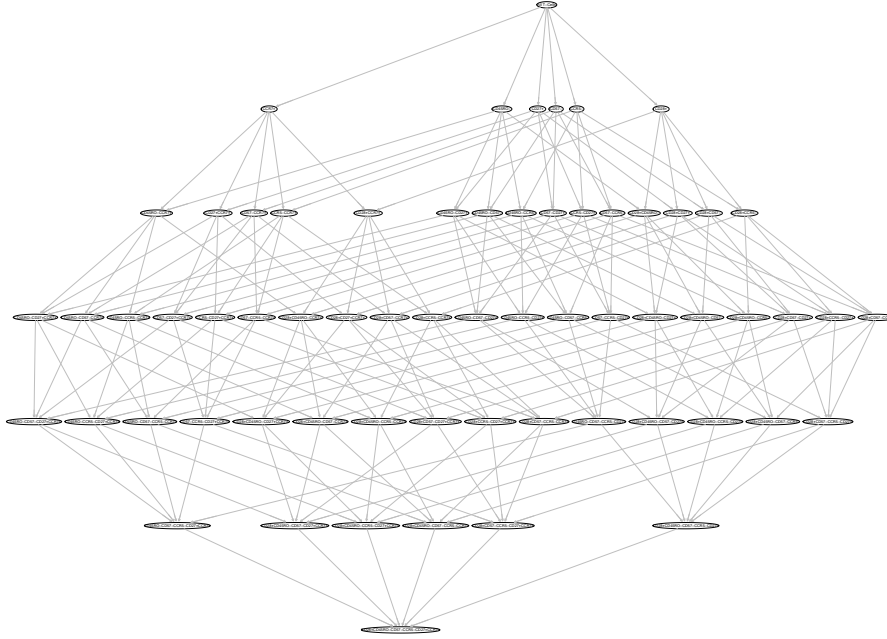


Figure 3.10: A complete cellular hierarchy for identifying naive T-cells. The colour of the nodes and the thickness of the edges have been removed to facilitate visualization of the complex graph.

To address this challenge, we developed RchyOptimyx, a computational tool that automatically characterizes the complex findings of high-dimensional exploratory FCM studies. RchyOptimyx sorts all parent populations of an immunophenotype of interest into hierarchies, and selects those hierarchies that

are better able to maintain the characteristics of the immunophenotype of interest (*e.g.*, correlation with a clinical outcome). This reveals the best order in which markers can be excluded from an immunophenotype. RchyOptimyx uses dynamic programming and efficient tools from graph theory to make the problem tractable using the computing resources readily available in most laboratories.

Since most cells can be described using more than one combination of markers, there usually are several alternative cellular hierarchies associated with every population. RchyOptimyx finds all these “paths” and merges them into a single hierarchy, starting from “all cells”, or any arbitrary point in a hierarchy, and finishing at the terminal population of interest. This reveals the relationships between different gating strategies and how they differ, and also facilitates the reproduction of high-dimensional exploratory studies using low-color instruments. The ability to suggest multiple panels is particularly important when designing new panels, because the choice of markers depends on a large number of external parameters including, but not limited to, reagents available through vendors, potential spectral overlaps, the instruments available, and budget limitations.

Another important use-case for RchyOptimyx is in the interpretation of the findings of bioinformatics pipelines. While these pipelines have recently been very successful in identifying cell populations correlated with clinical outcomes, their findings cannot be easily understood for two reasons: 1) they usually rely on high-dimensional clustering of the data and therefore cannot propose gating strategies for reproduction of their results; 2) their predictive power often relies on a large list of immunophenotypes. Some of these immunophenotypes are closely related (*e.g.*, refer to close or overlapping cell populations) while others are not. RchyOptimyx addresses the first problem by suggesting optimized gating hierarchies for identification of these cell populations to a desired level of purity or correlation with clinical outcome. The latter problem is addressed by summarizing closely related immunophenotypes using their most important common parents.

In evaluating RchyOptimyx, we combined its functionality with the automated gating functionality provided by flowMeans and flowType. However, RchyOptimyx can be built upon the results of any cell population identification method, including manual analysis, provided all intermediate cell populations (*i.e.*, each layer, removing one marker at a time) from the cell population of interest up to the desired start of the hierarchy are provided to the method.

We evaluated RchyOptimyx for three use-cases, using a small but high-dimensional mass cytometry dataset and a clinical dataset of high-dimensional conventional FCM assays of 466 patients, previously analyzed by both manual and automated analysis. First, we constructed cellular hierarchies for identification of cells that were produced in response to different stimulations. This use-case represents the problem of designing panels of surface markers (primarily for sorting) for cells that can only be defined using their intra-cellular signature (possibly after proper stimulation). For example, plasmacytoid dendritic cell (PDC)s are known to express the toll-like receptor 9 (TLR9) in response to

stimulation using CpG [55]. A large number of surface candidates were recently proposed for PDCs [63, 94, 86, 15]. An interesting direction to extend this work would be to measure all these markers in a single panel, subject to CpG stimulation (using appropriate controls) to design a panel of surface markers for PDCs. In this case, TLR9 could be used as the external variable for optimization.

Second, we demonstrated that RchyOptimyx can be used to simplify existing gating strategies, using as an example the identification of naive T-cells previously defined using a complex panel of six markers to a 95% purity using only three. This proof-of-concept use-case is relevant when a subset of markers needs to be selected for reproduction of the results using fewer colors. For certain biological use-cases, purity of higher than 95% can be required. For such use-cases, a larger number of markers for exclusion of non-naive T-cells should be included in the panel.

Third, we showed that RchyOptimyx, together with a complex bioinformatics pipeline, can analyze a large high-dimensional clinical dataset, to reveal correlates of a clinical outcome, hidden from previous manual and automated analysis of the same dataset. In addition, RchyOptimyx suggests the best gating strategies and marker panels for reproduction of these results in low-color settings. By identifying the best cellular hierarchies, RchyOptimyx allows the user to make an informed decision about the trade-off between the number of markers and the significance of the correlation with the clinical outcome. This feature is particularly important in hypothesis generating studies that need to be further validated using large clinical studies.

For the third example, it is important to note that the correct measure for the amount of correlation with a clinical outcome is an effect size (such as the root squared error of the estimated proportional hazard). However, such effect size does not provide any information about the significance of the correlation. As RchyOptimyx is intended to be a decision support tool, and in this case the decision is the degree to which a cell population can be generalized while maintaining the statistical significance of the correlation, we decided that the p-values of the log-rank tests were more appropriate for optimization of the hierarchies. To support this decision, we empirically investigated the differences between the p-values and effect sizes of the Cox proportional hazard models (Fig. 3.11) and concluded that these values are highly correlated (which is not surprising considering the large size of our cohort). It should be noted that as RchyOptimyx allows the user to choose which measure to provide, they can make this decision as appropriate for their specific data.

The concept of computationally extracting cellular hierarchies from FCM data has previously been introduced by the SPADE algorithm [10, 79]. SPADE generates a large number of multidimensional clusters and then connects them to each other using the distance between their mean/median fluorescence intensities. These are then manually annotated by biologists with domain knowledge. This makes SPADE useful for identification and visualization of a large number of clusters, particularly when expression of markers change gradually (*e.g.*, cell-cycle analysis and some intracellular studies). However, the hierarchies generated by SPADE are logically and conceptually different from those generated

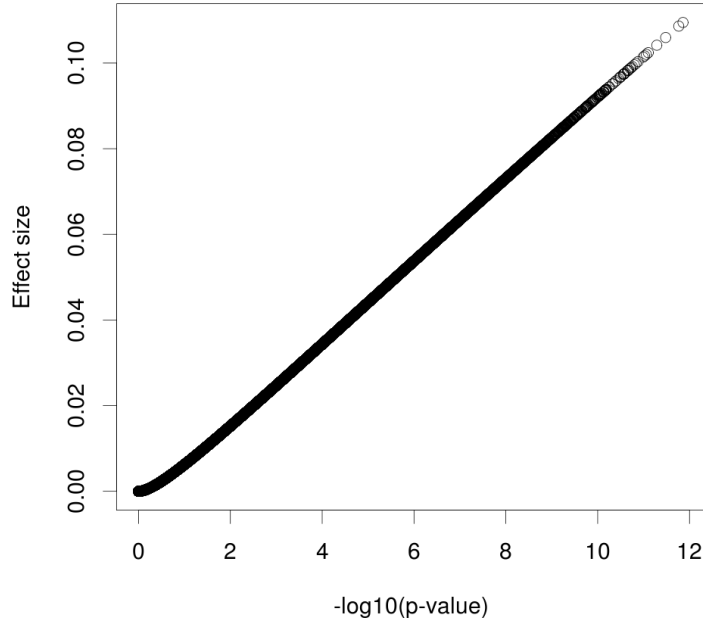


Figure 3.11: The correlation between the effect sizes and p-values of the log rank tests for the Cox proportional hazards models for each immunophenotype. The Pearson correlation coefficient was determined as 0.997, indicating a highly significant correlation with a p-value $< 2.2 \times 10^{-16}$.

by RchyOptimyx and have different use-cases. For example, the results of the mass cytometry dataset presented here are very close to results previously obtained from SPADE analysis. However, SPADE required manual annotation of the results by a human expert, using different plots demonstrating the expression of different surface markers and the intra-cellular marker of interest (Figure 2 and panel C of Figure 3 of [10]). More complicated relationships that involve several markers cannot be easily identified by these manual annotations. In addition, SPADE is limited in that the relationships between cell populations is exclusively defined using the multidimensional distances between them. However, two cell populations that are close to each other in the multidimensional space can be far in terms of specific markers (which can be the most important ones). The cellular hierarchies generated by RchyOptimyx are based on parent-child relationships, guided by an external variable (cell populations that have common parents with similar patterns of correlation with a clinical outcome or

intracellular response to stimulation are grouped together). This enables RchyOptimyx to automatically annotate a large number of cell populations identified by other methods (*e.g.*, manual gating or SPADE) in terms of the importance of the markers involved and summarize them in a single hierarchy.

There are several directions in which this work can be extended. RchyOptimyx provides no information about the robustness of the hierarchies. Bootstrapping strategies could be used to produce confidence intervals for the tree structure and increase generalizability to previously unseen data [93]. Also, our current implementation of RchyOptimyx assumes that every marker can be partitioned into a positive and negative population. While the underlying theory does support additional (*e.g.*, dim, bright, or low) populations, parts of the software package would need to be modified to accommodate these cases.

Availability

The RchyOptimyx R package (including source code, documentation, and examples) is freely available under an open source license (*Artistic 2.0*) and can be obtained from Bioconductor. The raw data and meta-data used in this study is publicly available through FlowRepository.org (under experiment ID *FR-FCM-ZZZK*) and through Cytobank.org (under experiment ID *6033*) for the PFC and CyTOF datasets, respectively.

3.4 Lymphoma Diagnosis Quality Checking

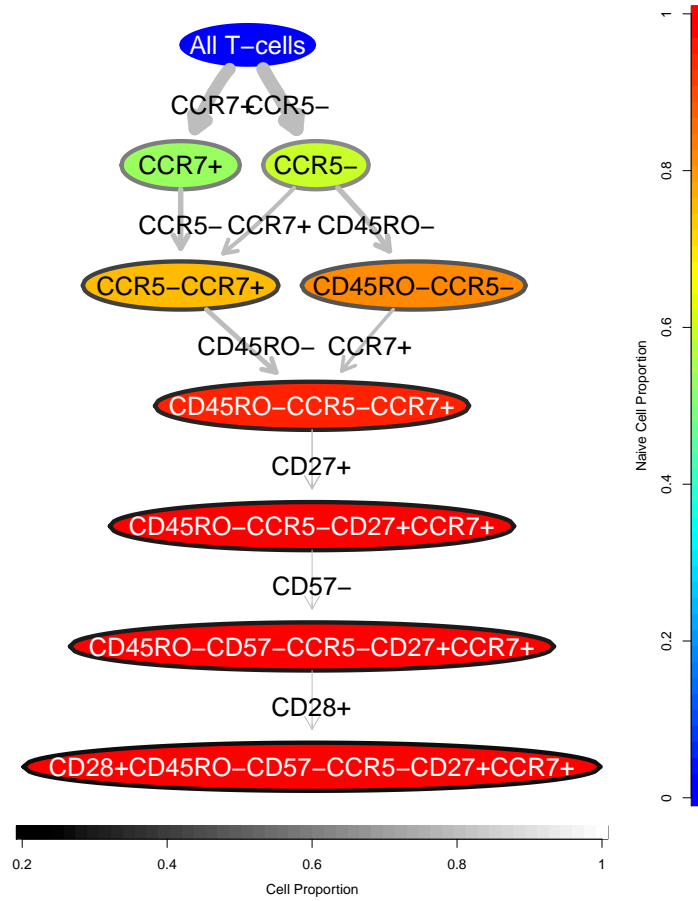


Figure 3.12: An optimized cellular hierarchy for identifying naive T-cells. The colour of the nodes and the thickness of the edges shows the purity and change in purity of the original naive phenotype within the given cell population, respectively. This is similar to Figure 6 in the main text except the color of the border of the nodes shows the cell proportion of the cell population.

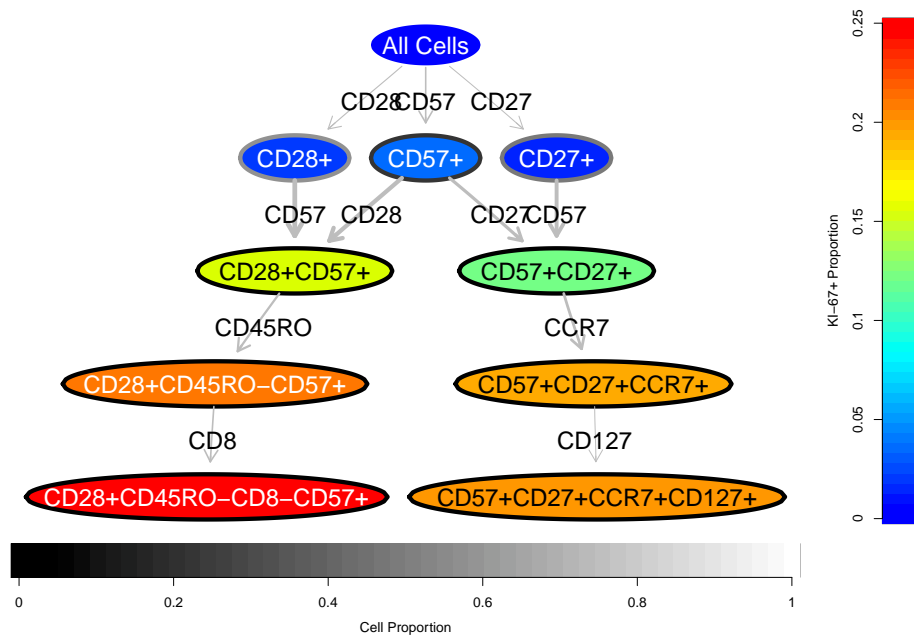


Figure 3.13: A cellular hierarchy for identifying KI-67⁺ T-cells using surface markers. The colour of the nodes and the thickness of the edges shows the proportion and change in proportion of KI-67⁺ T-cells, respectively. This is similar to Figure 7 in the main text except the color of the border of the nodes shows the cell proportion of the cell population.

“All models are wrong; some models are useful.”

- George Box

4

Adaptive Learning

Here we talk about adaptive and interpretable methods.

4.1 Challenges in Cancer Data

4.1.1 Cancer Heterogeneity

4.1.2 Batch Effects and Noise

4.2 RatBoost

4.2.1 Background

Over the past few decades, biology has become more targeted to individual cells, and has transformed into a high throughput research field, both in terms of the number of different measurement techniques, and the amount of variables measured by each technique (e.g., from Sanger sequencing to deep sequencing), and is more and more targeted to individual cells [88]. This has led to an unprecedented growth of biological information. Consequently, techniques that can help researchers find important insights into the data are becoming increasingly important. Predicting survival of cancer patients based on measurements from microarray experiments has been a field of great interest, but there is often very little overlap between the important genes or biomarkers identified by different studies [30]. Several reasons have been suggested to explain these findings (e.g., heterogeneity of cancer samples or insufficient sample size). Attempts have been made to incorporate additional information from other sources, such as protein-protein interaction (PPI) networks, to make the predictions more robust [23].

One of the latest approaches has integrated network and expression data by introducing a network-induced classification kernel (NICK) [58]. Although this method exhibits state-of-the-art performance, determining the variables leading to the method’s performance is not straightforward. Furthermore, many of these methods assume that all data is drawn from the same unknown underlying distribution. However this might not be the case for all measured genes in heterogeneous cancer samples.

In this work, we introduce a method that is aware of this potential bias. We utilize an estimate of the differences between observed and predicted values, during the generation of the final prediction method. Hence, we introduce a set of sparse classifiers based on $L1$ -SVMs [14], where each set of features used by one classifier is disjoint from the selected feature set of any other classifier. Furthermore, for each feature chosen by one of the classifiers, we introduce a regression model that uses additional features and is based on Gaussian process regression. These regression models are then used to estimate how predictable the features of each classifier are for each test sample, using other features of the test sample itself. This information is then used to find a confidence weighting of the classifiers, i.e. up-weighting classifiers with high confidence and down-weighting classifiers with lower confidence, for each test sample. Schapire and Singer have shown that incorporating confidences of classifiers can improve the performance of an ensemble method [85]. However, in their setting, confidences of classifiers are estimated using the training data and are thus fixed for all test samples, whereas in our setting, we estimate confidences of individual classifiers per given test sample. We show that this method exhibits state-of-the-art performance for different cancer types, with gene expression or methylation data sets as the input. Since the weighting of the classifiers is customized for each test sample, the estimated confidences can offer insights into the specific characteristics of each individual’s cancer. To facilitate interpretation of the model, we then create a visualization of the important genes found through this analysis for each test sample. Additionally, we show how the important genes of the training set can be found using our learning method and cross validation.

4.2.2 Methods

Materials

Data Sources: In this article, our method is applied to two different data types: gene expression data and DNA methylation data, which we retrieved from The Cancer Genome Atlas (TCGA) [95]. TCGA is a joint effort of the National Cancer Institute and the National Human Genome Research Institute to advance the understanding of the molecular basis of cancer. They provide access to the different measurements from cancer samples that have been analyzed to external researchers. Samples are categorized according to diagnosed cancer from which we use the following groups:

- *Acute Myeloid Leukemia (LAML)* [97]: At the time of writing, the data set includes 200 samples. 194 samples contain methylation data and we use

the part of the data measured by JHU-USC HumanMethylation450 arrays. 173 samples contain mRNA data measured by HG-U133 arrays. In this article the methylation data is referred to as TCGA-LAML. Among available characteristics of samples, “risk group” and “vital status” are chosen as target classes. These labels show the aggressiveness of the disease. In our analysis, regarding risk group, {favorable} and {intermediate/normal, poor} samples form our two group, and in the analysis of vital status, {alive} and {dead} samples form our two groups of samples.

- *Breast invasive carcinoma (BRCA)* [96]: This data set includes 993 samples with clinical data, and we use the methylation data component measured by JHU-USC HumanMethylation450 arrays. Only very few samples in this data set are indicated as having metastasized (8 samples). Hence the data are analyzed according to “tumor size”, “affected nearby lymph nodes”, “stage”, and “estrogen receptor”. Estrogen receptor was shown to be an important factor in prognosis [54], and along with other factors directly affects the decision for therapy [45, 67]. For tumor size {T1, T2} samples are one category and {T3, T4} the other category; in order to analyze affected nearby lymph nodes, {N0} is compared to {N1, N2, N3}; stage is analyzed as having {stage I, stage II} vs. {stage III} samples. Estrogen receptor status of samples is either positive or negative, and they form our two classes.

Data Preprocessing: To prepare gene expression data for analysis, microarray probes are mapped to their respective gene. If there are multiple probes for a gene, the median reported gene expression value of those probes is adopted as the gene expression for that gene.

Preparing the methylation data, we use the nearby gene for each methylation site available for each sample and each methylation site. The median beta value of methylation sites mapped to each gene is taken as the methylation value of the corresponding gene. In this process only methylation sites located on the promoter region of a gene are considered and others are discarded.

Learning a Mixture of Disjoint Classifiers

When dealing with cancer, we need to consider the fact that tumors of the same type of cancer can be very different in nature and they are usually classified as different cancer subtypes. In fact, even one single tumor can be very heterogeneous [49]. This means that the malignancies causing the cancer to happen are genetically different between subtypes, or even within subtypes, and it is possible to have multiple underlying cellular processes causing a particular cancer. Also it is important to note that the nature of our given data is such that the input features are properties measured from genes, e.g. gene expression or methylation values, and these variables are correlated and statistically dependent on each other. Our method tries to exploit these properties of the problem to infer an interpretable model with state-of-the-art performance.

Our method can be characterized by the following key parts:

Training phase:

- Fit several individual classifiers to the data, in such a way that the features of the data they use are disjoint sets.

Prediction phase:

- Calculate the prediction confidence of each individual classifier by:
 - Estimating the reliability of input features of the classifier;
 - Estimating the confidence of the output based on the decision values.
- Calculate a weighted prediction label based on the individual classifier confidences.

Properties of the Individual Classifiers: A wide variety of classifiers can be used within our framework. One requirement is that the classifier is regularized (i.e., the stronger the regularization, the less complex the model gets and consequently the less features are used). The classifier is also required to report the probability of its calculated output, or to give a decision value according to which it chooses the predicted class. We use an $L1$ regularized SVM for this purpose with a linear kernel [14]. The $L1$ regularization makes the SVM sparse, i.e. using only a few input features, and the linear kernel allows us to infer which features are used in the decision function of the SVM after it is fit to the data.

Training the Individual Classifiers: The model starts with no individual classifier and an empty set of excluded features. In each step, the excluded set of features is removed from the data, then a classifier is fit to the data. Next the features used by the most recent trained classifier are added to the excluded set. In the case of a linear kernel SVM, this is achieved by finding features with a non-zero coefficient in the model. This way the features being used by classifiers are disjoint and might represent different underlying causes of groups into which samples are to be classified.

Combining Classifiers by Estimating Confidences of Individual Predictors: Given a set of classifiers, the question is how to combine them to come up with a joint prediction value for each test sample for which we want to predict the output label. The intuition behind combining the classifiers is to put more weight on classifiers that use features whose behavior is similar to the training data. This is motivated by the fact that some parts of the test data might behave very differently to the training data, meaning that a classifier using these features should have lower performance than a classifier using features that are distributed similarly to the training data. Therefore we need to evaluate the reliability of the input features of each individual classifier. In scenarios like gene expression or methylation analysis, we usually have many input features. Furthermore, many features are correlated and statistically dependent. The idea of our new method is to build separate prediction models for each feature of each

classifier. These prediction models can then be used to obtain a confidence for the feature in a given test sample. These confidences can then be combined for each classifier to give a weighting of the classifiers for the given test sample. To evaluate an observed feature f , we try to choose a few statistically dependent features, and fit a model to predict f . To find these features, first the estimated mutual information of all other features with feature f is calculated [81]. Then, features having mutual information value within the top 5% or the 5 features with highest mutual information with f (if the top 5% features consist of less than 5 features), are selected as predictors of f . Given a test sample, the closer the predicted value of f is to the observed value, the more reliable it is. To quantify this, we need to not only know the predicted value of the feature, but also a confidence interval for that prediction. This can be achieved using Gaussian processes, which give the mean and variance of the posterior probability under the condition of observed values for selected features. A weighted average of these values gives us the overall reliability of the features of an individual classifier. A schematic view of the trained classifier is shown in Fig. 4.1.

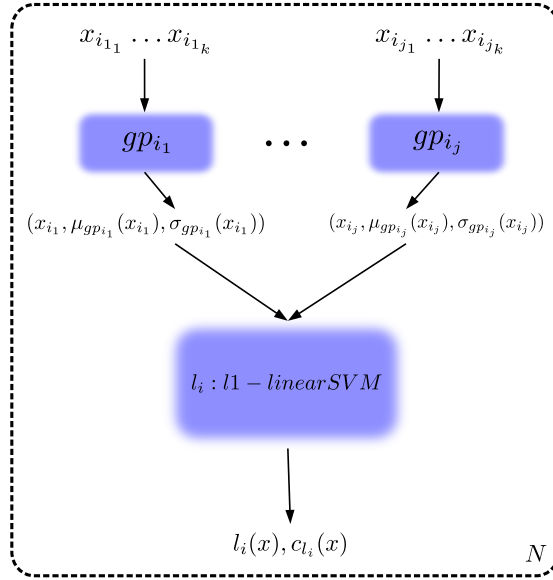


Figure 4.1: Schematic view of the method

In addition to the confidence in the classifier estimated by looking at the confidences of its individual features, we also account for the confidence that the classifier has in the prediction label of the test sample. If the method supplies such a confidence value (e.g., Gaussian processes), we can directly use it. Otherwise, we estimate it using the decision value. In our setting, the linear SVM gives a decision value whose sign defines the predicted class. Using these values we estimate a confidence for each individual classifier. Several approaches exist for deriving a confidence from the decision values [59]. Whether these or

other additional methods could lead to further improvements of our method, will be topic of further study.

More formally speaking, define X to be the set of input samples, X_s to be the input vector of sample s , y_s and \hat{y}_s to be respectively the original label and predicted output of sample s , Δ to be the set of individual classifiers, l_i to be an individual classifier, Φ_{l_i} the set of input features of classifier l_i , $l_i(X_s)$ to be the label predicted by classifier l_i for sample X_s , and f to be a feature, $X_{s,f}$ to be the observed value of feature f in sample X_s , $|w_{l_i}(f)|$ to be the absolute value of the weight of feature f in the decision function of classifier l_i , and g_f to be the Gaussian process predicting feature f using feature set Φ_f . Also $\mu_{g_f(X_s)}$ and $\sigma_{g_f(X_s)}$ are the mean and standard deviation of the posterior probability given by Gaussian process g_f under the condition of observing values of features in Φ_f , and μ_{l_i} and σ_{l_i} are respectively the expected mean and standard deviation of the decision value of classifier l_i . Here F is the cumulative distribution function of a standard normal distribution.

The training phase of the model is shown in Fig. 4.2, in which, N is the number of individual learners to be included in the model, Φ_l is the union over all Φ_{l_i} and $X_{-\Phi_l}$ is the input X after discarding all features of the set Φ_l . TOP is the function which selects the maximum of the top 5 and top 5% features f' of all features ordered by mutual information with feature f .

Now given a test sample X_s , the estimated confidence of a feature f is:

$$c_f(X_s) := 2 \cdot F \left(- \left| \frac{X_{s,f} - \mu_{g_f(X_s)}}{\sigma_{g_f(X_s)}} \right| \right) \quad (4.1)$$

Then the overall feature reliability or confidence of a classifier l_i is estimated as:

$$c_{l_i}^1(X_s) := \frac{\sum_{f \in \Phi_{l_i}} c_f(X_s) \cdot |w_{l_i}(f)|}{\sum_{f \in \Phi_{l_i}} |w_{l_i}(f)|} \quad (4.2)$$

Also the estimated output confidence of the classifier l_i is:

$$c_{l_i}^2(X_s) := 1 - 2 \cdot F \left(- \left| \frac{l_i(X_s) - \mu_{l_i}}{\sigma_{l_i}} \right| \right) \quad (4.3)$$

and the final confidence of the classifier l_i is then:

$$c_{l_i}(X_s) := c_{l_i}^1(X_s) \cdot c_{l_i}^2(X_s) \quad (4.4)$$

Finally, the predicted class \hat{y}_s is calculated as the sign of a weighted vote among individual classifiers:

$$\hat{y}_s := \text{sign} \left(\frac{\sum_{l_i \in \Delta} c_{l_i}(X_s) \cdot l_i(X_s)}{\sum_{l_i \in \Delta} c_{l_i}(X_s)} \right) \quad (4.5)$$

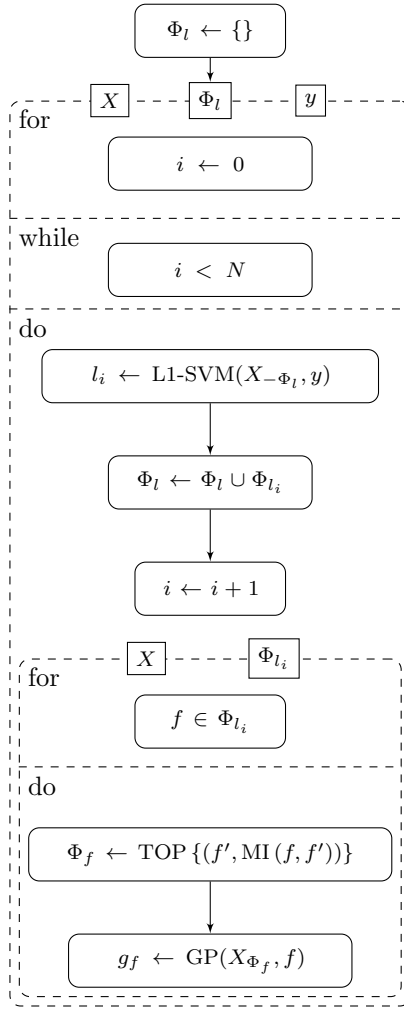


Figure 4.2: UML activity diagram of the training process

Visualization of Model Predictions

The interpretation of the model can be understood on two different ways. First we assume for a given training data set, the model is trained and a new test sample is given. For the given test sample it is possible to visualize the reliability of each used feature in individual classifiers, as well as the overall confidence of each individual classifier. Used features can be superimposed onto a PPI network as well as their reliability and the confidence of their respective individual classifier.

Gene expression and methylation level measurements from cancer samples are usually very noisy. Furthermore, cancers are usually very heterogeneous. Additionally, there might be different subgroups for each interesting group (e.g., cancer stage), for which the importance of the features also differs. To get a global picture of the important features, we therefore evaluate how often certain features are selected by the classifiers using 100 random train test partitionings with 80% of the data for training and 20% of the data for testing. To visualize high confidence relationships between features, we create a graph which has a node for every chosen feature in any of the 100 train partitions in any of the individual classifiers. The weight of an edge (s, t) is defined as the number of times the respective features have occurred together in an individual classifier. Then, all edges with low weights are discarded. In order to find a threshold to prune edges according to their weights, a Gaussian kernel density estimate is fit to the weights of the edges, and the threshold is chosen at the 90th percentile. Nodes that have an appearance frequency higher than the threshold are labeled by their gene names and edges having a higher weight than the threshold are kept in the graph.

For illustration purposes, choosing the regularization parameter is done in a way to maximize the number of genes selected with high confidence, as well as minimizing the number of genes pruned out in the process. It is important to remember that considering the results of the method under different regularization parameters is essential to make sure the selected genes possess a high confidence and are also stable regardless of sampling of the training data set.

Implementation Details

To compare the performance of our method with other methods, the implementations present in Python *scikit-learn(0.14)* package are taken. In the case of stochastic gradient boosting, the representing class is *GradientBoostingClassifier*, the number of classifiers is set to 100, and to make it sparse and prevent over-fitting, the maximum number of features for splits in trees is set to 5, and the maximum number of layers is set to 2. For AdaBoost, *AdaBoostClassifier* is used, which is an implementation of AdaBoost-SAMME [111], with weak learner set to *DecisionTreeClassifier* with maximum depth set to 2, and the number of weak classifiers set to 100. Parameters of the two boosting algorithms are chosen by a grid search on their parameter space over all the data sets and selecting the parameter sets which give a robust and stable result over all experiments.

As an SVM, ν -SVM with $\nu = 0.25$ is used, once with a linear kernel, and once with an RBF kernel; γ parameter of the RBF kernel is set to $(\text{num of features})^{-1}$. The ν parameter is set to the maximum value for which the optimization function is solvable with *libsvm* for all analyzed data sets [19]. Smaller values cause the SVM to overfit to the data and not generalize well. The Gaussian process’s correlation function is a squared-exponential, and mutual information is estimated using *minepy* package [6].

The PPI network used in our analysis is from the Human Protein Reference Database (HPRD) [74]. Almost all edges and relationships between proteins that are added to this database are manually extracted from literature by biologists, hence it has a lower rate of edges included in the database for which there is no evidence in the literature.

4.2.3 Results and discussion

Interpretability of Predictions

Here we present the results of running the method on the TCGA-LAML gene expression data set.

Visualization of Features Important for a Particular Test Sample:

Having a model trained on the data, and given a test sample, it is possible to infer and visualize which individual classifier(s) is (are) influencing the prediction most. To this end, individual learners as well as the features they use are visualized as in Fig. 4.3(a). In this figure, nodes with labels starting with “ L_- ” represent individual classifiers, and other nodes are labeled with their respective gene name. The color of the node shows its confidence compared to other nodes; the darker the node, the higher the confidence. In the case of a gene, it is the confidence or reliability of the feature (c_f), and in the case of an individual classifier, it is the overall estimated confidence (c_{l_i}). Edges show which classifier is using which genes in its decision function. The shape of a node represents the individual classifier they belong to.

To get a better overview of the individual features that were chosen by the classifiers for the particular test sample, we visualized the corresponding genes on a graph containing information about the PPI network in Fig. 4.3(b). We extracted the PPI information from HPRD as explained before. This way, it is possible to find over- or under-regulated pathways that might be responsible for the label (e.g., cancer stage) of the test sample. Since PPI networks can be quite dense, we removed parts of the induced network. For this purpose we computed each shortest path between all pairs of selected features. Then, the minimum spanning tree of that section was plotted, after removing branches with no selected feature.

Most of the features chosen by any of the classifiers (colored nodes) are not connected to any other chosen feature. It is known that there is in many cases a correlation between expression value of the genes whose corresponding proteins interact [50]. Therefore, a regularized model will only choose a subset of the correlated features. This explains the observation that features selected by a

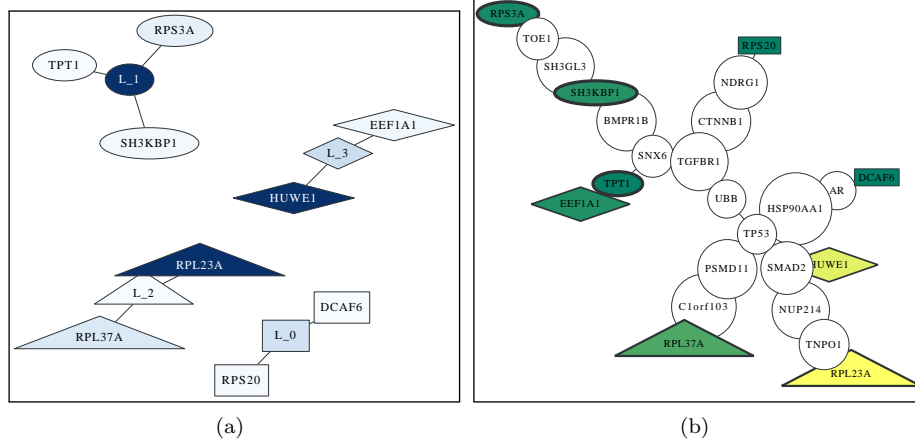


Figure 4.3: **Visualization of one model** A sample model for TCGA-LAML gene expression data **(a)** individual classifiers and their selected features; higher confidence of a node is shown by a darker color, **(b)** selected genes plotted over the PPI network; green and yellow show low and high confidence respectively, and the thickness of the border of the node shows the respective confidence of the individual classifier to which it belongs.

single model can be distant from each other on a PPI network; but if multiple disjoint sparse models are fit to the data, their selected features might happen to be close to each other on the PPI network (e.g., node TPT1 and node EEF1A1 in Fig. 4.3(b)).

It is worth noting that these plots are the result of analyzing one single given test sample. Therefore in practice, these interpretations can be used for each patient and if useful, influence the treatment that the oncologist prescribe for the patient.

Visualization of Important Global Features: As explained in Section 4.2.2, a graph is created from model structures of all 100 random training partitions, and then it is pruned to keep only high confidence nodes and edges. The density estimation of the graph edge weights and the pruned graph are plotted in Fig. 4.4 where the nodes with labels are the ones that are not pruned. The nodes in this figure that do not have any label, are the ones with frequency lower than the corresponding threshold. Among the features considered to be important were features that had previously been linked to leukemia such as SH3KBP1 [1].

What was more intriguing to see was that four out of the seven important features of the TCGA-LAML gene expression data set contained ribosomal proteins (when using the risk group label). For a long time ribosomes were just considered machines that perform an unbiased translation of genes from mRNA to amino acid sequences, but this view has recently been challenged [107]. One new hypothesis is that the ribosome introduces an additional regulatory layer.

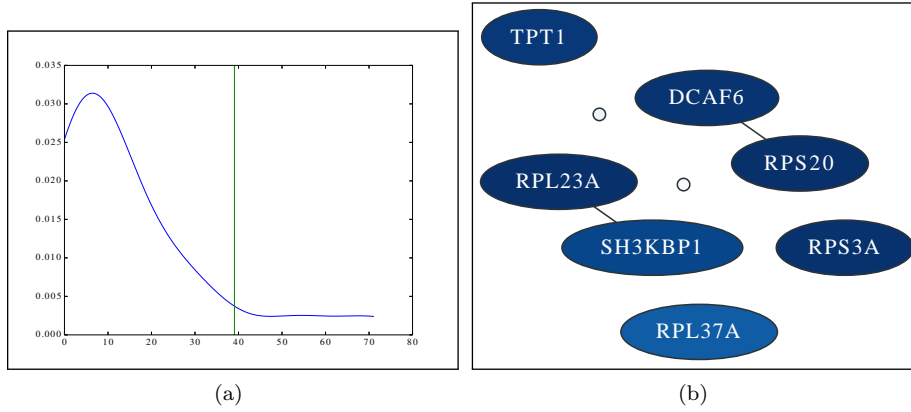


Figure 4.4: **(a) Determine pruning threshold** Threshold is determined by finding the point after which, 90% of the area under the curve is observed from left to right. The horizontal axis shows the observed frequency or weight of the edges. **(b) Important Global Features** High confidence nodes and edges of the graph generated from the model on TCGA-LAML gene expression data. Darker color represents higher rate of being selected by a classifier.

Therefore, it could very well be that mutations in ribosomal proteins can lead to a misregulation of expression levels of important genes and ultimately to the development of cancer (in this case leukemia). One of the ribosomal proteins we found was RPL23A. It has been shown that loss of RPL23A can impede growth and lead to morphological abnormalities in *Arabidopsis Thaliana* [107]. Therefore, a mutation in RPL23A might also have severe effects in humans. A missense mutation in RPL23A was recently found in patients having Diamond-Blackfan anemia, which is an inherited form of pure red cell aplasia (related to leukemia) [44]. Note that the model for LAML has low performance for the regularization value chosen. Nevertheless, the features shown here are also the ones with the highest confidence for models learnt with less regularization (with several other additional features). The models with less regularization show similar performance to the other methods shown in Fig. 4.5

Performance comparison

The performance of the method was compared with that of two ensemble methods, AdaBoost and stochastic gradient boosting, as well as an SVM with linear kernel, and an SVM with an RBF kernel. We also included our implementation of the NICK method [58]. We randomly partitioned the data into training and test sets with 80% of the data for training and 20% of the data for testing. To compare the performance of the different methods, Area Under the receiver operating characteristic Curve (AUC) [29] was calculated on the test set over the decision values returned by the methods on the individual samples. The

process was repeated 100 times to reduce random effects. As seen in Fig. 4.5, overall performances of all methods are comparable. In some cases a single SVM works better, in some other cases ensemble algorithms give a better performance. However, in most cases an improvement in performance is observed by adding individual learners to the model, with the greatest gains due to the first few individual learners added to the model.

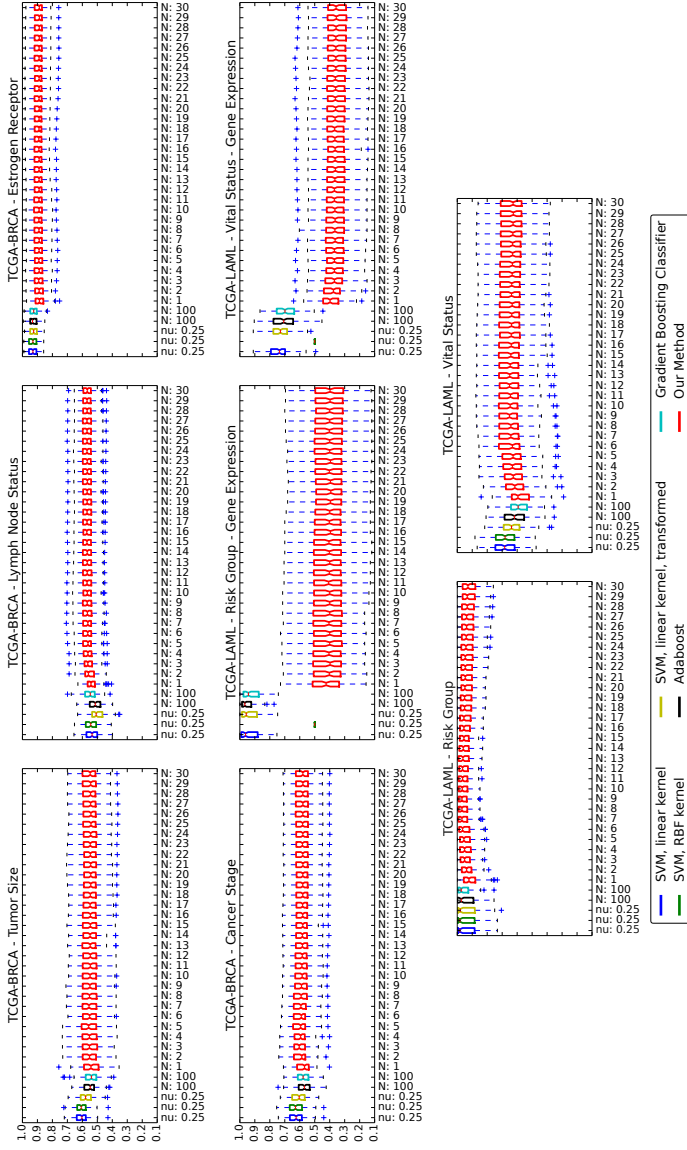


Figure 4.5: Performance Summary (AUC) Each box shows a 25–75% interval, as well as the median, which is shown as a horizontal line in each box.

4.2.4 Conclusions

Machine learning has become more and more popular in many real world scenarios for making sense of large collections of facts. Differences between the data used for training the method and new data for which the label should be predicted can limit the performance of prediction methods on those data. In this work we introduced a method that estimates these potential partial biases and incorporates them into the prediction function. We applied it to gene expression and DNA methylation measurements from cancer patients. Our method has state-of-the-art performance on many different prediction tasks. Furthermore, we show how to make sense of the predictions. Visualizing the important genes can lead to new biological insights, as shown for the TCGA-LAML data set with the risk group label. Instead of mapping the genes to PPI networks, they can be mapped to signaling pathways [53]. Recently, a study showed that most published signatures are not significantly more associated with cancer outcome than random signatures [100]. One of the reasons for this finding is that the data comes from slightly different underlying hidden data distributions. Since our new method estimates this bias and corrects for it by up-weighting the classifiers that have higher confidence, we expect that it should be less susceptible to such differences in the data.

4.3 Raccoon

5

Conclusion



RchyoOptimyx Appendix

Table A.1: The phenotypes with a high overlap with the BCR(pBLNK)⁺ compartment as identified by flowType. The table includes the cell proportion of these immunophenotypes (second column) and the differences in the cell proportion of BCR(pBLNK)⁺ cells in the stimulated and unstimulated assays (third column).

Phenotype Name	Cell Proportion	BCR ⁺ _(stim-unstim)
CD19+CD4-CD8-CD34+CD20+CD123+CD38-CD3-	0.001	0.160
CD19+CD4-CD34+CD20+CD123+CD38-CD3-	0.001	0.160
CD19+CD4-CD34+CD20+CD123+CD3-	0.001	0.155

Table A.2: The phenotypes with a high overlap with the IL7(pSTAT5)⁺ compartment as identified by flowType. The table includes the cell proportion of these immunophenotypes (second column) and differences in the cell proportion of IL7(pSTAT5)⁺ cells in the stimulated and unstimulated assays (third column).

Phenotype Name	Cell Proportion	IL7 ⁺ _(stim-unstim)
CD19-CD4+CD8+CD20+CD33+CD38-CD3+	0.008	0.364
CD19-CD4+CD8+CD20+CD33+CD3+	0.008	0.366
CD19-CD4+CD8+CD34+CD33+CD38-CD3+	0.008	0.366
CD19-CD4+CD8+CD34+CD33+CD3+	0.008	0.368
CD19-CD4+CD8+CD34+CD20+CD33+CD38-CD3+	0.006	0.399
CD19-CD4+CD8+CD34+CD20+CD33+CD3+	0.006	0.402
CD4+CD8+CD20+CD33+CD38-CD3+	0.011	0.365
CD4+CD8+CD20+CD33+CD3+	0.011	0.371
CD4+CD8+CD34+CD33+CD38-CD3+	0.011	0.366
CD4+CD8+CD34+CD33+CD3+	0.011	0.371
CD4+CD8+CD34+CD20+CD33+CD38-CD3+	0.008	0.399
CD4+CD8+CD34+CD20+CD33+CD3+	0.009	0.405
CD19+CD4+CD8+CD20+CD33+CD38-CD3+	0.003	0.364
CD19+CD4+CD8+CD20+CD33+CD3+	0.003	0.378
CD19+CD4+CD8+CD34+CD33+CD38-CD3+	0.003	0.359
CD19+CD4+CD8+CD34+CD33+CD3+	0.003	0.372
CD19+CD4+CD8+CD34+CD20+CD33+CD38-CD3+	0.002	0.397
CD19+CD4+CD8+CD34+CD20+CD33+CD3+	0.002	0.409

Table A.3: The phenotypes with a high overlap with the LPS(p-p38)⁺ compartment as identified by flowType. The table includes the cell proportion of these immunophenotypes (second column) and differences in the cell proportion of LPS(p-p38)⁺ cells in the stimulated and unstimulated assays (third column).

Phenotype Name	Cell Proportion	LPS ⁺ _(stim-unstim)
CD19-CD4-CD8-CD34-CD20-CD33+CD123-CD38-CD3-	0.008	0.474
CD19-CD4-CD8-CD34-CD20-CD33+CD123-CD3-	0.008	0.473
CD19-CD4-CD8-CD34-CD20-CD33+CD38-CD3-	0.009	0.466
CD19-CD4-CD8-CD34-CD20-CD33+CD3-	0.009	0.465
CD19-CD4-CD8-CD34-CD33+CD123-CD38-CD3-	0.022	0.460
CD19-CD4-CD8-CD34-CD33+CD123-CD3-	0.022	0.459
CD19-CD4-CD8-CD34-CD33+CD38-CD3-	0.022	0.452
CD19-CD4-CD8-CD34-CD33+CD3-	0.022	0.451
CD19-CD4-CD8-CD34-CD20+CD33+CD123-CD38-CD3-	0.013	0.450
CD19-CD4-CD8-CD34-CD20+CD33+CD123-CD3-	0.013	0.449
CD19-CD4-CD8-CD20-CD33+CD123-CD38-CD3-	0.023	0.453
CD19-CD4-CD8-CD20-CD33+CD123-CD3-	0.023	0.452
CD19-CD4-CD34-CD20-CD33+CD123-CD38-CD3-	0.011	0.456
CD19-CD4-CD34-CD20-CD33+CD123-CD3-	0.011	0.455
CD19-CD8-CD34-CD20-CD33+CD123-CD38-CD3-	0.012	0.462
CD19-CD8-CD34-CD20-CD33+CD123-CD3-	0.012	0.461
CD19-CD8-CD34-CD20-CD33+CD38-CD3-	0.012	0.454
CD19-CD8-CD34-CD20-CD33+CD3-	0.012	0.454
CD4-CD8-CD34-CD20-CD33+CD123-CD38-CD3-	0.011	0.462
CD4-CD8-CD34-CD20-CD33+CD123-CD3-	0.011	0.461
CD4-CD8-CD34-CD20-CD33+CD38-CD3-	0.011	0.454
CD4-CD8-CD34-CD20-CD33+CD3-	0.011	0.454
CD8-CD34-CD20-CD33+CD123-CD38-CD3-	0.015	0.450
CD8-CD34-CD20-CD33+CD123-CD3-	0.015	0.449

Table A.4: Statistically significant immunophenotypic correlates of survival of HIV⁺ subjects are predicted by flowType. The p-values of the log rank tests, 95% confidence intervals calculated using bootstrapping, adjusted p-values using Bonferroni's method, coefficients and R^2 values of the Cox proportional hazards regression models, and the frequency of the cells are provided as columns of the table.

#	Phenotype	p-value	p-value, CI	adjusted p-value	CPHR Coefficient	R^2	Cell Frequency
1	CD28-CD45RO+CD57-CCR5+	5.3e-07	(4.3e-14, 1.3e-02)	2e-02	20.5	0.056	0.03048
2	CD28-CD8+CD57-CD127-	2.9e-07	(2.3e-14, 3.8e-04)	1e-02	12.3	0.060	0.05975
3	CD28-CD45RO+CD57-CCR7-	5.1e-07	(2.3e-14, 6.1e-04)	2e-02	15.7	0.057	0.03829
4	CD28-CD45RO+CD4-CD57-	3.9e-07	(2.3e-14, 1.1e-03)	1e-02	13.2	0.058	0.04357
5	CD45RO+CD4-CD57-CD127-	2.7e-07	(1.2e-13, 7.1e-03)	1e-02	12.8	0.059	0.05062
6	CD28-CD45RO+CD57-CD127-	4.7e-08	(1.7e-14, 6.8e-04)	2e-03	16.0	0.067	0.03732
7	CD45RO+CD4-CD27-CD127-	4.4e-07	(5.8e-14, 1.1e-03)	2e-02	14.3	0.057	0.04830
8	CD28-CD45RO+CD57-	5.6e-07	(4.4e-14, 4.1e-04)	2e-02	12.4	0.056	0.05015
9	CD45RO+CD4-CD127-	6.9e-07	(4.7e-15, 2.9e-03)	2e-02	9.6	0.056	0.07176
10	CD28-CD45RO+CD4-CD127-	3.1e-07	(0.0e+00, 5.7e-03)	1e-02	11.7	0.059	0.05300
11	CD28-CD45RO+CD57-CCR5+CD27-CCR7+CD127-	4.7e-07	(5.7e-14, 7.7e-03)	2e-02	171.4	0.057	0.00315
12	CD28-CD45RO+CD4-CD57-CCR5+CD27-CCR7+CD127-	4.9e-07	(1.8e-13, 3.9e-04)	2e-02	176.2	0.057	0.00294
13	CD28-CD57-CD127-	3.3e-07	(3.4e-15, 8.0e-03)	1e-02	8.0	0.058	0.12341
14	CD28-CD4-CD57-	8.8e-07	(2.2e-15, 2.9e-03)	3e-02	7.2	0.054	0.15525
15	CD4-CD57-CD27-CD127-	6.2e-08	(2.4e-14, 4.7e-03)	2e-03	9.5	0.065	0.12173
16	CD28-CD57-CCR7-CD127-	4.7e-08	(4.2e-14, 3.3e-03)	2e-03	9.7	0.067	0.09721
17	CD28-CD57-CCR7-CD127-	2.8e-07	(9.7e-15, 1.0e-02)	1e-02	9.8	0.059	0.08417
18	CD28-CD4-CD57-CD127-	3.3e-08	(2.0e-12, 5.7e-04)	1e-03	9.1	0.068	0.10852
19	CD4-CD57-CCR7-CD127-	6.9e-07	(3.8e-15, 2.3e-03)	2e-02	8.8	0.056	0.09501
20	CD45RO-CD4-CD57+CCR5-CD27+CCR7-CD127-	6.1e-07	(1.2e-12, 2.6e-03)	2e-02	498.4	0.056	0.00097
21	CD28-CD45RO-CD4-CD57+CCR5-CD27+CCR7-CD127-	2.9e-07	(0.0e+00, 7.7e-03)	1e-02	561.2	0.060	0.00074
22	CD45RO-CD8+CD57+CCR5-CD27+CCR7-CD127-	1.2e-07	(4.6e-14, 3.3e-04)	5e-03	638.6	0.063	0.00068
23	CD45RO-CD8+CD4-CD57+CCR5-CD27+CCR7-CD127-	1.2e-07	(5.1e-14, 2.0e-03)	5e-03	638.6	0.063	0.00068
24	CD28-CD45RO-CD4-CD57+CCR5-CD27+CCR7-CD127-	5.7e-07	(1.1e-13, 2.3e-03)	2e-02	298.3	0.056	0.00099
25	CD28-CD28-CCR5+	1.0e-11	(2.9e-13, 2.8e-03)	4e-07	96.1	0.101	0.00547
26	KI-67+CD28-CCR5+CD27-	8.7e-12	(1.5e-14, 8.9e-04)	3e-07	115.3	0.102	0.00453
27	KI-67+CCR5+	1.3e-11	(2.4e-14, 7.0e-03)	5e-07	53.4	0.100	0.01192
28	KI-67+CD28+CD45RO+CD57-CCR7-CD127-	4.2e-09	(5.6e-16, 3.0e-03)	2e-04	241.3	0.077	0.00209
29	KI-67+CD45RO-CD4-CD27-CCR7-CD127-	1.2e-09	(2.0e-14, 4.4e-03)	4e-05	161.9	0.082	0.00297
30	KI-67+CD28-CD45RO-CD8-CD4-	5.0e-09	(2.9e-12, 1.7e-03)	2e-04	176.0	0.076	0.00225
31	KI-67+CD8-CD4-	8.1e-09	(6.1e-13, 4.5e-02)	3e-04	58.1	0.074	0.00738
32	KI-67+CCR5+CD27-CCR7-	2.0e-11	(3.8e-14, 6.0e-04)	8e-07	109.8	0.099	0.00532
33	KI-67+CD28-CCR5+CCR7-	1.3e-10	(3.1e-13, 2.0e-03)	5e-06	147.3	0.091	0.00392
34	KI-67+CD28-CD8-CCR5+CCR7+CD127-	2.6e-09	(1.6e-14, 1.1e-02)	1e-04	625.8	0.079	0.00061
35	KI-67+CD28+CD45RO+CD8+CD57-CD27+CCR7+	6.7e-07	(3.8e-13, 1.5e-03)	3e-02	585.4	0.055	0.00051
36	KI-67+CD28+CD45RO+CD8+CD4-CD57-CD27+CCR7+	6.7e-07	(1.1e-16, 4.7e-03)	3e-02	585.4	0.055	0.00051
37	KI-67+CD8+CD27-CCR7-CD127-	4.7e-11	(1.3e-13, 1.4e-03)	2e-06	141.3	0.095	0.00292
38	KI-67+CD8+CD4-CD27-CCR7-CD127-	4.7e-11	(1.3e-13, 1.3e-03)	2e-06	141.3	0.095	0.00292
39	KI-67+CD28-CD8+CD27-CCR7-CD127-	2.7e-11	(1.0e-13, 7.6e-04)	1e-06	164.5	0.097	0.00241
40	KI-67+CD28-CD8+CD4-CD27-CCR7-CD127-	2.7e-11	(2.7e-13, 1.4e-03)	1e-06	164.5	0.097	0.00241

Table A.4: Statistically significant immunophenotypic correlates of survival of HIV⁺ subjects are predicted by flowType. The p-values of the log rank tests, 95% confidence intervals calculated using bootstrapping, adjusted p-values using Bonferroni's method, coefficients and R^2 values of the Cox proportional hazards regression models, and the frequency of the cells are provided as columns of the table.

#	Phenotype	p-value	p-value, CI	adjusted p-value	CPHR Coefficient	R^2	Cell Frequency
41	KI-67+CD28-CD8+CCR7-CD127-	6.6e-11	(5.6e-14, 1.5e-02)	3e-06	132.9	0.094	0.00293
42	KI-67+CD28-CD8+CD4-CCR7-CD127-	6.6e-11	(1.2e-14, 8.4e-04)	3e-06	132.9	0.094	0.00293
43	KI-67+CD45RO+CD8+CD27-CCR7-	1.2e-09	(4.0e-12, 2.8e-03)	5e-05	143.6	0.082	0.00216
44	KI-67+CD45RO+CD8+CD4-CD27-CCR7-	1.2e-09	(1.0e-12, 1.2e-02)	5e-05	143.6	0.082	0.00216
45	KI-67+CD28-CD45RO+CD8+CD27-CCR7-	1.0e-09	(1.9e-15, 7.3e-04)	4e-05	188.5	0.082	0.00155
46	KI-67+CD28-CD45RO+CD8+CD4-CD27-CCR7-	1.0e-09	(1.7e-13, 2.0e-03)	4e-05	188.5	0.082	0.00155
47	KI-67+CD45RO+CD8+CD27-CD127-	7.1e-10	(1.2e-14, 6.8e-03)	3e-05	152.4	0.084	0.00221
48	KI-67+CD45RO+CD8+CD4-CD27-CD127-	7.1e-10	(3.4e-14, 1.5e-03)	3e-05	152.4	0.084	0.00221
49	KI-67+CD28-CD45RO+CD8+CD27-CD127-	5.0e-10	(6.0e-13, 3.1e-03)	2e-05	201.3	0.085	0.00163
50	KI-67+CD28-CD45RO+CD8+CD4-CD27-CD127-	5.0e-10	(4.6e-14, 2.7e-03)	2e-05	201.3	0.085	0.00163
51	KI-67+CD28-CD45RO+CD8+CD4-CD127-	1.0e-09	(1.2e-15, 3.2e-03)	4e-05	150.5	0.083	0.00222
52	KI-67+CD45RO+CD8+CD4-CD127-	2.2e-09	(1.5e-11, 3.6e-03)	4e-05	150.5	0.083	0.00222
53	KI-67+CD28-CD45RO+CD8+CD4-CCR7-	8.0e-09	(2.8e-13, 2.1e-03)	9e-05	99.8	0.079	0.00362
54	KI-67+CD28-CD45RO+CD8+CD4-CCR7-	8.0e-09	(2.7e-12, 7.2e-04)	3e-04	133.6	0.074	0.00209
55	KI-67+CD28-CD45RO+CD57-CCR7+CD127-	5.9e-08	(4.0e-15, 4.5e-03)	2e-03	376.6	0.066	0.00075
56	KI-67+CD28-CD45RO+CD4-CD57-CCR7+CD127-	5.9e-08	(4.8e-13, 3.9e-03)	2e-03	409.6	0.066	0.00070
57	KI-67+CD57-CD27-CD127-	5.9e-10	(3.2e-14, 2.7e-03)	2e-05	44.9	0.085	0.00806
58	KI-67+CD28-CD27-CD127-	4.8e-10	(7.3e-15, 2.5e-03)	2e-05	50.6	0.086	0.00711
59	KI-67+CD4-CD127-	1.3e-10	(4.4e-16, 9.7e-03)	5e-06	37.1	0.091	0.01159
60	KI-67+CD28-CD127-	4.9e-10	(1.1e-12, 1.4e-03)	2e-05	41.4	0.086	0.00823
61	KI-67+CD4-CD27-	5.6e-09	(2.1e-14, 2.6e-03)	2e-04	28.6	0.075	0.01122
62	KI-67+CD28-CD4-CD27-	1.8e-09	(3.6e-13, 5.3e-03)	7e-05	40.2	0.080	0.00785
63	KI-67+CD27-CD127-	1.3e-09	(9.8e-15, 1.1e-03)	5e-05	33.0	0.082	0.01052
64	KI-67+CCR7-CD127-	6.5e-11	(1.4e-15, 9.6e-04)	2e-06	47.3	0.094	0.00947
65	KI-67+CD4-CD27-CCR7-	9.6e-11	(1.1e-16, 1.5e-03)	4e-06	52.1	0.092	0.00764
66	KI-67+CD4-CCR7-	1.7e-10	(3.0e-14, 1.0e-02)	7e-06	41.4	0.090	0.00987
67	KI-67+CD45RO+CD57-CCR7-	1.4e-09	(6.6e-13, 1.2e-03)	5e-05	49.6	0.081	0.00695
68	KI-67+CD45RO+CD57-CD27-CCR7-	9.1e-10	(8.0e-12, 2.5e-03)	3e-05	66.4	0.083	0.00505
69	KI-67+CD28-CD45RO+	2.0e-09	(8.0e-13, 2.5e-03)	3e-05	45.3	0.080	0.00851
70	KI-67+CD28-CD45RO+	1.3e-09	(1.2e-12, 2.4e-03)	5e-04	54.9	0.072	0.00525
71	KI-67+CD45RO+CD127-	1.1e-09	(4.4e-16, 1.5e-02)	4e-05	42.5	0.082	0.00834
72	KI-67+CD45RO+CD57-CD127-	2.9e-10	(1.5e-14, 6.4e-04)	1e-05	55.0	0.088	0.00719
73	KI-67+CD28-CD45RO+CD8+CD27-	9.2e-09	(2.6e-15, 2.3e-03)	4e-04	138.0	0.073	0.00201
74	KI-67+CD28-CD45RO+CD8+CD4-CD27-	1.9e-09	(1.0e-15, 4.6e-03)	4e-04	138.0	0.073	0.00201
75	KI-67+CD8+CD4-CD57-CD27-CD127-	1.9e-09	(5.9e-14, 7.0e-03)	7e-05	113.8	0.080	0.00274
76	KI-67+CD28-CD45RO+CD8+	9.3e-09	(5.9e-13, 1.4e-03)	4e-04	102.7	0.073	0.00279
77	KI-67+CD28-CD45RO+CD8+CD4-	2.1e-08	(0.0e+00, 1.6e-03)	4e-04	102.7	0.073	0.00279
78	KI-67+CD45RO+CD8+	3.0e-08	(6.9e-15, 6.8e-04)	8e-04	59.1	0.070	0.00512
79	KI-67+CD8+CCR7-	2.1e-08	(7.7e-13, 2.8e-03)	1e-03	49.5	0.068	0.00530
80	KI-67+CD8+CD27-CCR7-	8.3e-09	(1.0e-13, 3.6e-03)	3e-04	70.7	0.074	0.00377

Table A.4: Statistically significant immunophenotypic correlates of survival of HIV⁺ subjects are predicted by flowType. The p-values of the log rank tests, 95% confidence intervals calculated using bootstrapping, adjusted p-values using Bonferroni's method, coefficients and R^2 values of the Cox proportional hazards regression models, and the frequency of the cells are provided as columns of the table.

#	Phenotype	p-value	p-value, CI	adjusted p-value	CPHR Coefficient	R^2	Cell Frequency
81	KI-67+CD4-	2.8e-08	(1.0e-13, 2.3e-03)	1e-03	17.1	0.069	0.01627
82	KI-67+CD28-CD4-	1.1e-08	(5.9e-14, 4.0e-03)	4e-04	26.7	0.073	0.00950
83	KI-67+CD127-	2.7e-08	(1.2e-12, 2.1e-03)	1e-03	19.1	0.069	0.01460
84	KI-67+CCR7-	8.4e-08	(3.4e-15, 2.3e-03)	3e-03	18.3	0.064	0.01311
85	KI-67+CD27-CCR7-	3.5e-08	(1.7e-13, 1.2e-03)	1e-03	25.2	0.068	0.00998
86	KI-67+CD45RO+CD27-	7.5e-07	(5.4e-13, 1.8e-03)	3e-02	24.0	0.055	0.00862
87	KI-67+CD45RO+CD57-	1.2e-07	(2.1e-13, 3.1e-03)	5e-03	22.9	0.062	0.01123
88	KI-67+CD4-CD57-	1.3e-08	(3.8e-15, 2.1e-03)	5e-04	25.3	0.072	0.01209
89	KI-67+CD28-CD4-CD57-	9.7e-09	(5.5e-12, 1.2e-03)	4e-04	37.7	0.073	0.00698
90	KI-67+CD57-CD127-	3.3e-09	(1.3e-13, 3.3e-03)	1e-04	28.1	0.078	0.01128
91	KI-67+CD45RO+CCR7-	4.2e-09	(7.8e-15, 2.5e-03)	2e-04	37.5	0.077	0.00819
92	KI-67+CD57-CCR7-	2.7e-08	(2.8e-13, 2.8e-03)	1e-03	26.6	0.069	0.01008
93	KI-67+CD57-CD27-CCR7-	1.2e-08	(4.9e-13, 2.6e-03)	5e-04	36.8	0.072	0.00762
94	KI-67+CD28-CCR7-	3.3e-09	(4.6e-14, 5.7e-03)	1e-04	37.7	0.078	0.00739
95	KI-67+CD28-CD27-CCR7-	3.3e-09	(2.6e-14, 6.5e-04)	1e-04	43.0	0.078	0.00647
96	KI-67+CD28-	1.9e-07	(4.0e-15, 2.7e-03)	7e-03	18.3	0.061	0.01053
97	KI-67+CD28-CD27-	7.1e-08	(1.5e-12, 8.6e-04)	3e-03	26.3	0.065	0.00874
98	KI-67+CD28-CD8-	8.3e-08	(5.5e-14, 2.5e-03)	3e-03	44.2	0.064	0.00523
99	KI-67+CD45RO+	8.9e-07	(1.9e-13, 2.5e-03)	3e-02	15.4	0.054	0.01343
100	KI-67+CD8+CD57-	1.1e-06	(4.4e-14, 3.1e-03)	4e-02	28.3	0.053	0.00648
101	KI-67+CD8+CD27-	6.4e-07	(2.3e-14, 1.1e-02)	2e-02	35.2	0.056	0.00560

List of Tables

2.1	An example number of samples and features in our usual data . .	4
A.1	The phenotypes with a high overlap with the BCR(pBLNK) ⁺ compartment as identified by flowType. The table includes the cell proportion of these immunophenotypes (second column) and the differences in the cell proportion of BCR(pBLNK) ⁺ cells in the stimulated and unstimulated assays (third column).	56
A.2	The phenotypes with a high overlap with the IL7(pSTAT5) ⁺ compartment as identified by flowType. The table includes the cell proportion of these immunophenotypes (second column) and differences in the cell proportion of IL7(pSTAT5) ⁺ cells in the stimulated and unstimulated assays (third column).	56
A.3	The phenotypes with a high overlap with the LPS(p-p38) ⁺ compartment as identified by flowType. The table includes the cell proportion of these immunophenotypes (second column) and differences in the cell proportion of LPS(p-p38) ⁺ cells in the stimulated and unstimulated assays (third column).	57
A.4	Statistically significant immunophenotypic correlates of survival of HIV ⁺ subjects are predicted by flowType. The p-values of the log rank tests, 95% confidence intervals calculated using bootstrapping, adjusted p-values using Bonferroni's method, coefficients and R^2 values of the Cox proportional hazards regression models, and the frequency of the cells are provided as columns of the table.	58
A.4	Statistically significant immunophenotypic correlates of survival of HIV ⁺ subjects are predicted by flowType. The p-values of the log rank tests, 95% confidence intervals calculated using bootstrapping, adjusted p-values using Bonferroni's method, coefficients and R^2 values of the Cox proportional hazards regression models, and the frequency of the cells are provided as columns of the table.	59

A.4	Statistically significant immunophenotypic correlates of survival of HIV ⁺ subjects are predicted by flowType. The p-values of the log rank tests, 95% confidence intervals calculated using bootstrapping, adjusted p-values using Bonferroni's method, coefficients and R^2 values of the Cox proportional hazards regression models, and the frequency of the cells are provided as columns of the table.	60
-----	---	----

List of Figures

2.1	Illustration of the optimal hyperplane in a support vector machine model, for a 2-dimensional data.	7
2.2	A given weighted directed graph and the highlighted shortest path between vertices A and F	10
3.1	[DIRECT QUOTE] Fluorescence emission spectra for FITC and PE. The emission spectrum (the wavelengths of light generated by excitation of these molecules) is shown for an excitation at $488nm$ (the same as the argon-ion laser line). FITC emission is maximal at $\sim 515nm$; typically, a filter centered on $530nm$ is used to collect the emitted light (shaded region). The emission of is farther red, with a maximum at $\sim 575nm$; typically, a filter centered on this emission maximum is used to collect. Note that PE has some emission in the wavelength bands used to collect PE fluorescence (B); typically, the amount of light in the $575nm$ band is $\sim 15\%$ of that in the $530nm$ band (A). The PE has very little emission in the $530nm$ band (C), usually less than 2% of the emission in the $575nm$ band (D) [82].	14
3.2	a-b. Run time comparison of flowType-DP to flowType-BF in terms of number of cells (a) and number of markers (b). c-d. Possible thresholds for marker combinations using flowType-DP for typical mass cytometry data (c) and polychromatic flow cytometry data (d). e-f. Three/four-partition flowType-generated, RchyOptimyx-visualized cell type hierarchy on a bone marrow sample from a patient with AML. Cell population identification strategy used for SSC and CD45, with the CD34-enriched subset highlighted (e). RchyOptimyx analysis showing CD34 enrichment (f).	17

3.3	A complete cellular hierarchy for prediction of HIV's clinical outcome using $\text{KI67}^+\text{CD4}^-\text{CCR5}^+\text{CD127}^-$ T-cells. The color of the nodes indicates the significance of the correlation with clinical outcome (p-value of the logrank test for the Cox proportional hazards model) and the width of each edge (arrow) shows the amount of change in this variable between the respective nodes. The positive and negative correlation of each immunophenotype with outcome can be seen from the arrow type leading to the node; however, as all correlations are negative in this hierarchy, only one arrow type is shown.	21
3.4	Dynamic programming algorithm for two cell populations defined by 3 markers. The best path for each of the cell population is shown in red and blue respectively. As an example, the red path ends at $\text{CD4}^+\text{CCR5}^+\text{CD127}^+$. Three markers are available to be added. First, CD4 is added (changes from does not matter to positive). Then two options will be available for the next step (CD127 and CCR5). After selection of CCR5, only one option will be left for the final step (CD127). Therefore for three markers, $\frac{3 \cdot (3-1)}{2} = 6$ comparisons were required. Left: A hierarchy for the two paths. The label of an edge is the name of the single marker phenotype that is the difference between its head set (s) and its tail set (t). Right: the dynamic programming space for the 3 markers. Black spheres mark the nodes in the dynamic programming space used by the two paths. The colors of the nodes on the left match that of the square tori on the right and correspond to the relative score of each cell population.	23
3.5	An optimized cellular hierarchy for prediction of HIV's clinical outcome using $\text{KI67}^+\text{CD4}^-\text{CCR5}^+\text{CD127}^-$ T-cells. The color of the nodes shows the significance of the correlation with clinical outcome (p-value of the logrank test for the Cox proportional hazards model) and the width of each edge (arrow) shows the amount of change in this variable between the respective nodes.	25
3.6	All immunophenotypes ordered by their overlap with the cell population of interest. The red dashed lines indicate the cutoffs used for selecting the immunophenotypes with "high overlap".	26
3.7	Three optimized hierarchies for identification of cell populations with maximum response to IL7, BCR, and LPS measured by pSTAT5, pBLNK, and p-p38, respectively. The colour of the nodes and the thickness of the edges shows the proportion and change in proportion of cells expressing the intracellular marker of interest, respectively.	27
3.8	An optimized cellular hierarchy for identifying naive T-cells. The color of the nodes and the thickness of the edges shows the purity and change in purity of the original naive phenotype within the given cell population, respectively.	28

3.9	An optimized hierarchy for all three populations correlated with protection against HIV. The color of the nodes shows the significance of the correlation with the clinical outcome (p-value of the logrank test for the Cox proportional hazards model) and the width of each edge (arrow) shows the amount of change in this variable between the respective nodes. The positive and negative correlation of each immunophenotype with outcome can be seen from the arrow type leading to the node; however as all correlations are negative in this hierarchy, only one arrow type is shown.	30
3.10	A complete cellular hierarchy for identifying naive T-cells. The colour of the nodes and the thickness of the edges have been removed to facilitate visualization of the complex graph.	31
3.11	The correlation between the effect sizes and p-values of the log rank tests for the Cox proportional hazards models for each immunophenotype. The Pearson correlation coefficient was determined as 0.997, indicating a highly significant correlation with a p-value $< 2.2 \times 10^{-16}$	34
3.12	An optimized cellular hierarchy for identifying naive T-cells. The colour of the nodes and the thickness of the edges shows the purity and change in purity of the original naive phenotype within the given cell population, respectively. This is similar to Figure 6 in the main text except the color of the border of the nodes shows the cell proportion of the cell population.	36
3.13	A cellular hierarchy for identifying KI-67 ⁺ T-cells using surface markers. The colour of the nodes and the thickness of the edges shows the proportion and change in proportion of KI-67 ⁺ T-cells, respectively. This is similar to Figure 7 in the main text except the color of the border of the nodes shows the cell proportion of the cell population.	37
4.1	Schematic view of the method	43
4.2	UML activity diagram of the training process	45
4.3	Visualization of one model A sample model for TCGA-LAML gene expression data (a) individual classifiers and their selected features; higher confidence of a node is shown by a darker color, (b) selected genes plotted over the PPI network; green and yellow show low and high confidence respectively, and the thickness of the border of the node shows the respective confidence of the individual classifier to which it belongs.	48

4.4	(a) Determine pruning threshold Threshold is determined by finding the point after which, 90% of the area under the curve is observed from left to right. The horizontal axis shows the observed frequency or weight of the edges. (b) Important Global Features High confidence nodes and edges of the graph generated from the model on TCGA-LAML gene expression data. Darker color represents higher rate of being selected by a classifier.	49
4.5	Performance Summary (AUC) Each box shows a 25–75% interval, as well as the median, which is shown as a horizontal line in each box.	51

Bibliography

- [1] J. Adélaïde, V. Gelsi-Boyer, J. Rocquain, N. Carbuccion, D. J. Birnbaum, P. Finetti, F. Bertucci, M. J. Mozziconacci, N. Vey, D. Birnbaum, and M. Chaffanet. Gain of CBL-interacting protein, a possible alternative to CBL mutations in myeloid malignancies. *Leukemia*, 24(8):1539–41, Aug. 2010.
- [2] N. Aghaeepour, P. K. Chattopadhyay, A. Ganesan, K. O’Neill, H. Zare, A. Jalali, H. H. Hoos, M. Roederer, and R. R. Brinkman. Early immunologic correlates of HIV protection can be identified from computational analysis of complex multivariate T-cell flow cytometry assays. *Bioinformatics*, 28(7):1009–1016, 2012.
- [3] N. Aghaeepour, G. Finak, H. Hoos, T. R. Mosmann, R. Brinkman, R. Gottardo, R. H. Scheuermann, et al. Critical assessment of automated flow cytometry data analysis techniques. *Nature Methods*, 10(3):228–238, 2013.
- [4] N. Aghaeepour, A. Jalali, K. O’Neill, P. K. Chattopadhyay, M. Roederer, H. H. Hoos, and R. R. Brinkman. RchyOptimyx: Cellular hierarchy optimization for flow cytometry. *Cytometry Part A*, 81(12):1022–30, 2012.
- [5] N. Aghaeepour, R. Nikolic, H. Hoos, and R. Brinkman. Rapid cell population identification in flow cytometry data. *Cytometry Part A*, 79(1):6–13, 2011.
- [6] D. Albanese, M. Filosi, R. Visintainer, S. Riccadonna, G. Jurman, and C. Furlanello. minerva and minepy: a c engine for the mine suite and its r, python and matlab wrappers. *Bioinformatics*, 29(3):407–408, 2013.
- [7] A. Bashashati, N. Johnson, A. Khodabakhshi, M. Whiteside, H. Zare, D. Scott, K. Lo, R. Gottardo, F. Brinkman, J. Connors, et al. B cells with high side scatter parameter by flow cytometry correlate with inferior survival in diffuse large b-cell lymphoma. *American Journal of Clinical Pathology*, 137(5):805–814, 2012.
- [8] R. Bellman. On a routing problem. Technical report, DTIC Document, 1956.
- [9] S. Bendall, G. Nolan, M. Roederer, and P. Chattopadhyay. A deep profiler’s guide to cytometry. *Trends in Immunology*, 33(7):323–332, 2012.

- [10] S. Bendall, E. Simonds, P. Qiu, E. Amir, P. Krutzik, R. Finck, R. Brugnier, R. Melamed, A. Trejo, O. Ornatsky, et al. Single-cell mass cytometry of differential immune and drug responses across a human hematopoietic continuum. *Science*, 332(6030):687–696, 2011.
- [11] S. C. Bendall, G. P. Nolan, M. Roederer, and P. K. Chattopadhyay. A deep profiler’s guide to cytometry. *Trends in Immunology*, 33(7):323–332, July 2012.
- [12] A. Biancotto, P. Dagur, J. Chris Fuchs, M. Langweiler, and J. Philip McCoy Jr. OMIP-004: In-depth characterization of human T regulatory cells. *Cytometry Part A*, 81(1):360–361, 2011.
- [13] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.
- [14] P. S. Bradley and O. L. Mangasarian. Feature selection via concave minimization and support vector machines. In *ICML*, volume 98, pages 82–90, 1998.
- [15] W. Cao. Molecular characterization of human plasmacytoid dendritic cells. *Journal of Clinical Immunology*, 29(3):257–264, 2009.
- [16] K. Castro, J. Ward, L. Slutsker, J. Buehler, H. Jaffe, R. Berkelman, and J. Curran. Revised classification system for HIV infection and expanded surveillance case definition for AIDS among adolescents and adults. *MMWR Recomm Rep*, 41:1–19, 1992.
- [17] C. Chan, F. Feng, J. Ottinger, D. Foster, M. West, and T. Kepler. Statistical mixture modeling for cell subtype identification in flow cytometry. *Cytometry Part A*, 73(8):693–701, 2008.
- [18] C. Chan, L. Lin, J. Frelinger, V. Hébert, D. Gagnon, C. Landry, R. Sékaly, J. Enzor, J. Staats, K. Weinhold, et al. Optimization of a highly standardized carboxyfluorescein succinimidyl ester flow cytometry panel and gating strategy design using discriminative information measure evaluation. *Cytometry Part A*, 77(12):1126–1136, 2010.
- [19] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.
- [20] P. Chattopadhyay and M. Roederer. Cytometry: Today’s technology and tomorrow’s horizons. *Methods*, 57(3):251–258, Feb 2012.
- [21] P. Chattopadhyay, M. Roederer, and D. Price. OMIP-002: Phenotypic analysis of specific class I multimers for any of four epitopes. *Cytometry Part A*, 77(9):821–822, 2010.

- [22] P. K. Chattopadhyay, C. M. Hogerkorp, and M. Roederer. A chromatic explosion: the development and future of multiparameter flow cytometry. *Immunology*, 125(4):441, 2008.
- [23] H.-Y. Chuang, E. Lee, Y.-T. Liu, D. Lee, and T. Ideker. Network-based classification of breast cancer metastasis. *Molecular systems biology*, 3:140, Jan. 2007.
- [24] E. Costa, C. Pedreira, S. Barrena, Q. Lecrevisse, J. Flores, S. Quijano, J. Almeida, M. del Carmen García-Macias, S. Bottcher, J. Van Dongen, et al. Automated pattern-guided principal component analysis vs expert-based immunophenotypic classification of b-cell chronic lymphoproliferative disorders: a step forward in the standardization of clinical immunophenotyping. *Leukemia*, 24(11):1927–1933, 2010.
- [25] D. R. Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 215–242, 1958.
- [26] F. Craig, R. Brinkman, E. Ten, and N. Aghaeepour. Computational analysis optimizes the flow cytometric evaluation for lymphoma. *Cytometry Part B - Clinical Cytometry*, Digital preprint, 2013.
- [27] T. G. Dietterich. Ensemble learning. *The handbook of brain theory and neural networks*, pages 405–408, 2002.
- [28] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.
- [29] J. P. Egan. Signal detection theory and ROC analysis. *Academic Press, New York*, 1975.
- [30] L. Ein-Dor, I. Kela, G. Getz, D. Givol, and E. Domany. Outcome signature genes in breast cancer: is there a unique set? *Bioinformatics*, 21(2):171–8, Jan. 2005.
- [31] M. Eller and J. Currier. OMIP-007: Phenotypic analysis of human natural killer cells. *Cytometry Part A*, 81(6):447–449, 2012.
- [32] D. Eppstein. Finding the k shortest paths. *SIAM Journal on computing*, 28(2):652–673, 1998.
- [33] G. Finak, A. Bashashati, R. Brinkman, and R. Gottardo. Merging mixture components for cell population identification in flow cytometry. *Advances in Bioinformatics*, v09, 2009.
- [34] G. Finak, J.-M. Perez, A. Weng, and R. Gottardo. Optimizing transformations for automated, high throughput analysis of flow cytometry data. *BMC bioinformatics*, 11(1):546, 2010.
- [35] R. W. Floyd. Algorithm 97: shortest path. *Communications of the ACM*, 5(6):345, 1962.

- [36] K. Foulds, M. Donaldson, and M. Roederer. OMIP-005: Quality and phenotype of antigen-responsive rhesus macaque T cells. *Cytometry Part A*, 81(6):360–361, 2012.
- [37] G. N. Frederickson. An optimal algorithm for selection in a min-heap. *Information and Computation*, 104(2):197–214, 1993.
- [38] M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM (JACM)*, 34(3):596–615, 1987.
- [39] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. System Sci.*, 55(1):119–139, 1997.
- [40] J. H. Friedman. Stochastic gradient boosting. *Comput. Stat. Data Anal.*, 38(4):367–378, 2002.
- [41] M. J. Fulwyler. Electronic separation of biological cells by volume. *Science*, 150(3698):910–911, 1965.
- [42] A. Ganesan, P. K. Chattopadhyay, T. M. Brodie, J. Qin, W. Gu, J. R. Mascola, N. L. Michael, D. A. Follmann, M. Roederer, C. Decker, T. Whitman, S. Tasker, A. Weintrob, G. Wortmann, M. Zapor, M. Landrum, V. Marconi, J. Okulicz, N. Crum-Cianflone, M. Bavaro, H. Chun, R. V. Barthel, A. Johnson, B. Agan, N. Aronson, W. Bradley, G. Gandits, L. Jagodzinski, R. O’Connell, C. Eggleston, and J. Powers. Immunologic and virologic events in early HIV infection predict subsequent rate of progression. *Journal of Infectious Diseases*, 201:272–284, Jan 2010.
- [43] L. Gattinoni, E. Lugli, Y. Ji, Z. Pos, C. Paulos, M. Quigley, J. Almeida, E. Gostick, Z. Yu, C. Carpenito, et al. A human memory t cell subset with stem cell-like properties. *Nature Medicine*, pages 1290–1297, 2011.
- [44] H. T. Gazda, M. Preti, M. R. Sheen, M.-F. O’Donohue, A. Vlachos, and S. M. Davies et al. Frameshift mutation in p53 regulator RPL26 is associated with multiple physical abnormalities and a specific pre-ribosomal RNA processing defect in diamond-blackfan anemia. *Human mutation*, 33(7):1037–44, July 2012.
- [45] A. Goldhirsch, J. H. Glick, R. D. Gelber, A. S. Coates, and H.-J. Senn. Meeting highlights: international consensus panel on the treatment of primary breast cancer. *Journal of Clinical Oncology*, 19(18):3817–3827, 2001.
- [46] S. Gordon, B. Cervasi, P. Odorizzi, R. Silverman, F. Aberra, G. Ginsberg, J. Estes, M. Paiardini, I. Frank, and G. Silvestri. Disruption of intestinal CD4+ T cell homeostasis is a key marker of systemic CD4+ T cell activation in HIV-infected individuals. *The Journal of Immunology*, 185(9):5169–5179, 2010.

- [47] F. Hahne, A. H. Khodabakhshi, A. Bashashati, C.-J. Wong, R. D. Gascoyne, A. P. Weng, V. Seyfert-Margolis, K. Bourcier, A. Asare, T. Lumley, et al. Per-channel basis normalization methods for flow cytometry data. *Cytometry Part A*, 77(2):121–131, 2010.
- [48] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.
- [49] G. H. Heppner. Tumor heterogeneity. *Cancer research*, 44(6):2259–2265, 1984.
- [50] R. Jansen, D. Greenbaum, and M. Gerstein. Relating whole-genome expression data with protein-protein interactions. *Genome research*, 12(1):37–46, 2002.
- [51] H. Jaspán, L. Liebenberg, W. Hanekom, W. Burgers, D. Coetzee, A. Williamson, F. Little, L. Myer, R. Coombs, D. Sodora, et al. Immune activation in the female genital tract during hiv infection predicts mucosal cd4 depletion and hiv shedding. *Journal of Infectious Diseases*, 204(10):1550–1556, 2011.
- [52] V. M. Jiménez and A. Marzal. A lazy version of eppsteins k shortest paths algorithm. In *Experimental and Efficient Algorithms*, pages 179–191. Springer, 2003.
- [53] M. Kanehisa, S. Goto, Y. Sato, M. Kawashima, M. Furumichi, and M. Tanabe. Data, information, knowledge and principle: back to metabolism in KEGG. *Nucleic acids research*, 42(Database issue):D199–205, Jan. 2014.
- [54] W. A. Knight, R. B. Livingston, E. J. Gregory, and W. L. McGuire. Estrogen receptor as an independent prognostic factor for early recurrence in breast cancer. *Cancer research*, 37(12):4669–4671, 1977.
- [55] A. Krug, A. Towarowski, S. Britsch, S. Rothenfusser, V. Hornung, R. Bals, T. Giese, H. Engelmann, S. Endres, A. Krieg, et al. Toll-like receptor expression reveals CpG DNA as a unique microbial stimulus for plasmacytoid dendritic cells which synergizes with CD40 ligand to induce high amounts of IL-12. *European Journal of Immunology*, 31(10):3026–3037, 2001.
- [56] H.-T. KUHN. Aw (1951) nonlinear programming. In *2nd Berkeley Symposium*. Berkeley, University of California Press.
- [57] L. Lamoreaux, R. Koup, and M. Roederer. OMIP-009: Characterization of antigen-specific human T-cells. *Cytometry Part A*, 81(5):362–363, 2012.
- [58] O. Lavi, G. Dror, and R. Shamir. Network-induced classification kernels for gene expression profile analysis. *J Comput Biol.*, 19(6):694–709, June 2012.

- [59] H.-T. Lin, C.-J. Lin, and R. C. Weng. A note on Platt’s probabilistic outputs for support vector machines. *Machine Learning*, 68(3):267–276, Aug. 2007.
- [60] K. Lo, R. Brinkman, and R. Gottardo. Automated gating of flow cytometry data via robust model-based clustering. *Cytometry Part A*, 73(4):321–332, 2008.
- [61] H. T. Maecker, J. P. McCoy, and R. Nussenblatt. Standardizing immunophenotyping for the Human Immunology Project. *Nature Reviews Immunology*, 12:191–200, 2012.
- [62] Y. Mahnke and M. Roederer. OMIP-001: Quality and phenotype of Ag-responsive human T-cells. *Cytometry Part A*, 77(9):819–820, 2010.
- [63] T. Marafioti, J. Paterson, E. Ballabio, K. Reichard, S. Tedoldi, K. Hollowood, M. Dictor, M. Hansmann, S. Pileri, M. Dyer, et al. Novel markers of normal and neoplastic human plasmacytoid dendritic cells. *Blood*, 111(7):3778–3792, 2008.
- [64] J. Mercer. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical transactions of the royal society of London. Series A, containing papers of a mathematical or physical character*, pages 415–446, 1909.
- [65] S. Mukherjee. *The emperor of all maladies: a biography of cancer*. Simon and Schuster, 2011.
- [66] D. Murdoch, J. Staats, and K. Weinhold. OMIP-006: Phenotypic subset analysis of human T regulatory cells via polychromatic flow cytometry. *Cytometry Part A*, 81(4):281–283, 2012.
- [67] National Institutes of Health Consensus Development Panel and others. National institutes of health consensus development conference statement: adjuvant therapy for breast cancer, november 1—3, 2000. *J Natl Cancer Inst.*, 93(13):979–989, 2001.
- [68] U. Naumann, G. Luta, and M. Wand. The curvHDR method for gating flow cytometry samples. *BMC Bioinformatics*, 11(1):11–44, 2010.
- [69] O. Ornatsky, D. Bandura, V. Baranov, M. Nitz, M. Winnik, and S. Tanner. Highly multiparametric analysis by mass cytometry. *Journal of Immunological Methods*, 361(6030):1–20, 2010.
- [70] M. Y. Park and T. Hastie. L1-regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):659–677, 2007.
- [71] D. R. Parks, M. Roederer, and W. A. Moore. A new logic display method avoids deceptive effects of logarithmic scaling for low signals and compensated data. *Cytometry Part A*, 69(6):541–551, 2006.

- [72] K. Pearson. Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, pages 240–242, 1895.
- [73] S. Perfetto, P. Chattopadhyay, and M. Roederer. Seventeen-colour flow cytometry: unravelling the immune system. *Nature Reviews Immunology*, 4(8):648–655, 2004.
- [74] S. Peri, J. D. Navarro, R. Amanchy, T. Z. Kristiansen, C. K. Jonnalagadda, V. Surendranath, V. Niranjana, B. Muthusamy, T. Gandhi, M. Gronborg, et al. Development of human protein reference database as an initial platform for approaching systems biology in humans. *Genome research*, 13(10):2363–2371, 2003.
- [75] F. Preijers, E. Huys, and B. Moshaver. OMIP-010: A new 10-color monoclonal antibody panel for polychromatic immunophenotyping of small hematopoietic cell samples. *Cytometry Part A*, 81(6):453–455, 2012.
- [76] S. Pyne, X. Hu, K. Wang, E. Rossin, T. Lin, L. Maier, C. Baecher-Allan, G. McLachlan, P. Tamayo, D. Hafler, et al. Automated high-dimensional flow cytometric data analysis. *Proceedings of the National Academy of Sciences*, 106(21):8519–8524, 2009.
- [77] Y. Qian, Y. Liu, J. Campbell, E. Thomson, Y. M. Kong, and R. H. Scheuermann. Fcstrans: an open source software system for fcs file conversion and data transformation. *Cytometry Part A*, 81(5):353–356, 2012.
- [78] Y. Qian, C. Wei, F. Eun-Hyung Lee, J. Campbell, J. Halliley, J. Lee, J. Cai, Y. Kong, E. Sadat, E. Thomson, et al. Elucidation of seventeen human peripheral blood B-cell subsets and quantification of the tetanus response using a density-based method for the automated identification of cell populations in multidimensional flow cytometry data. *Cytometry Part B: Clinical Cytometry*, 78(S1):S69–S82, 2010.
- [79] P. Qiu, E. Simonds, S. Bendall, K. Gibbs Jr, R. Bruggner, M. Linderman, K. Sachs, G. Nolan, and S. Plevritis. Extracting a cellular hierarchy from high-dimensional cytometry data with spade. *Nature Biotechnology*, 29:886–891, 2011.
- [80] C. E. Rasmussen. Gaussian processes for machine learning. 2006.
- [81] D. N. Reshef, Y. A. Reshef, H. K. Finucane, S. R. Grossman, G. McVean, P. J. Turnbaugh, E. S. Lander, M. Mitzenmacher, and P. C. Sabeti. Detecting novel associations in large data sets. *Science*, 334(6062):1518–1524, 2011.
- [82] M. Roederer. Compensation in flow cytometry. *Current Protocols in Cytometry*, pages 1–14, 2002.

- [83] M. Roederer, J. Nozzi, and M. Nason. Spice: Exploration and analysis of post-cytometric complex multivariate datasets. *Cytometry Part A*, 79(2):167–174, 2011.
- [84] M. Roederer and A. Tárnok. OMIPsOrchestrating multiplexity in polychromatic science. *Cytometry Part A*, 77(9):811–812, 2010.
- [85] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine learning*, 37(3):297–336, 1999.
- [86] P. Schuster, N. Donhauser, K. Pritschet, M. Ries, S. Haupt, N. Kittan, K. Korn, and B. Schmidt. Co-ordinated regulation of plasmacytoid dendritic cell surface receptors upon stimulation with herpes simplex virus type 1. *Immunology*, 129(2):234–247, 2010.
- [87] C. E. Shannon. The mathematical theory of communication. 1963. *MD computing: computers in medical practice*, 14(4):306–317, 1996.
- [88] E. Shapiro, T. Biezuner, and S. Linnarsson. Single-cell sequencing-based technologies will revolutionize whole-organism science. *Nat Rev Genet.*, 14(9):618–30, Sept. 2013.
- [89] H. M. Shapiro. *Practical flow cytometry*. John Wiley & Sons, 2005.
- [90] A. Smola and V. Vapnik. Support vector regression machines. *Advances in neural information processing systems*, 9:155–161, 1997.
- [91] A. J. Smola and B. Schölkopf. *Learning with kernels*. Citeseer, 1998.
- [92] I. Sugár and S. Sealfon. Misty Mountain clustering: application to fast unsupervised flow cytometry gating. *BMC Bioinformatics*, 11(1):502–508, 2010.
- [93] R. Suzuki and H. Shimodaira. Pvclust: an r package for assessing the uncertainty in hierarchical clustering. *Bioinformatics*, 22(12):1540–1542, 2006.
- [94] M. Swiecki and M. Colonna. Unraveling the functions of plasmacytoid dendritic cells during viral infections, autoimmunity, and tolerance. *Immunological Reviews*, 234(1):142–162, 2010.
- [95] The Cancer Genome Atlas Network. The Cancer Genome Atlas (TCGA). <https://tcga-data.nci.nih.gov/tcga/>, 2006.
- [96] The Cancer Genome Atlas Network. Comprehensive molecular portraits of human breast tumours. *Nature*, 490(7418):61–70, 2012.
- [97] The Cancer Genome Atlas Network. Genomic and epigenomic landscapes of adult *de novo* acute myeloid leukemia. *N Engl J Med.*, 368(22):2059–2074, 2013.

- [98] V. Vapnik. *The nature of statistical learning theory*. Springer Science & Business Media, 2013.
- [99] V. Vapnik and A. Chervonenkis. A note on one class of perceptrons. *Automation and remote control*, 25(1), 1964.
- [100] D. Venet, J. E. Dumont, and V. Detours. Most random gene expression signatures are significantly associated with breast cancer outcome. *PLoS computational biology*, 7(10):e1002240, Oct. 2011.
- [101] J. P. Vial and F. Lacombe. Immunophenotyping of acute leukemia: Utility of CD45 for blast cell identification. *Methods in Cell Biology*, 64:343–358, 2001.
- [102] F. Villanova, P. D. Meglio, M. Inokumad, N. Aghaeepour, E. Perucha, J. Mollon, L. Nomura, M. Hernandez-Fuentes, A. Copeh, T. Prevosti, S. Heck, V. Maino, G. Lord, R. R. Brinkman, , and F. O. Nestle. Integration of lyoplate based flow cytometry and computational analysis for standardized immunological biomarker discovery. *PLoS ONE*, 8(7), 2013.
- [103] S. H. Walker and D. B. Duncan. Estimation of the probability of an event as a function of several independent variables. *Biometrika*, 54(1-2):167–179, 1967.
- [104] C. Wei, J. Jung, and I. Sanz. OMIP-003: Phenotypic analysis of human memory B cells. *Cytometry Part A*, 79(11):894–896, 2011.
- [105] A. Weintrob, A. Fieberg, B. Agan, A. Ganesan, N. Crum-Cianflone, V. Marconi, M. Roediger, S. Fraser, S. Wegner, and G. Wortmann. Increasing age at HIV seroconversion from 18 to 40 years is associated with favorable virologic and immunologic responses to HAART. *JAIDS Journal of Acquired Immune Deficiency Syndromes*, 49(1):40–47, 2008.
- [106] A. K. White, M. VanInsberghe, O. I. Petriv, M. Hamidi, D. Sikorski, M. A. Marra, J. Piret, S. Aparicio, and C. L. Hansen. High-throughput microfluidic single-cell RT-qPCR. *Proceedings of the National Academy of Sciences*, 108(34), Aug. 2011.
- [107] S. Xue and M. Barna. Specialized ribosomes: a new frontier in gene regulation and organismal biology. *Nat Rev Mol Cell Biol.*, 13(6):355–69, June 2012.
- [108] J. Y. Yen. Finding the k shortest loopless paths in a network. *management Science*, 17(11):712–716, 1971.
- [109] H. Zare, A. Bashashati, R. Kridel, N. Aghaeepour, G. Haffari, J. Connors, R. Gascoyne, A. Gupta, R. Brinkman, and A. Weng. Automated analysis of multidimensional flow cytometry data improves diagnostic accuracy between mantle cell lymphoma and small lymphocytic lymphoma. *American Journal of Clinical Pathology*, 137(1):75–85, 2012.

- [110] H. Zare, P. Shooshtari, A. Gupta, and R. Brinkman. Data reduction for spectral clustering to analyze high throughput flow cytometry data. *BMC Bioinformatics*, 11(1):403–413, 2010.
- [111] J. Zhu, H. Zou, S. Rosset, and T. Hastie. Multi-class AdaBoost. *Statistics and its Interface*, 2(3), 2009.
- [112] C. Zuleger and M. Albertini. OMIP-008: Measurement of Th1 and Th2 cytokine polyfunctionality of human T cells. *Cytometry Part A*, 81(6):450–452, 2012.