Statistical File Matching of Flow Cytometry Data

Gyemin Lee[a,1,2,*], William Finn[b], Clayton Scott[a,c,1]

[a] *Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, 48109 USA*
[b] *Department of Pathology, University of Michigan, Ann Arbor, MI, 48109 USA*
[c] *Department of Statistics, University of Michigan, Ann Arbor, MI, 48109 USA*

Abstract

Flow cytometry is a technology that rapidly measures antigen-based markers associated to cells in a cell population. Although analysis of flow cytometry data has traditionally considered one or two markers at a time, there has been increasing interest in multidimensional analysis. However, flow cytometers are limited in the number of markers they can jointly observe, which is typically a fraction of the number of markers of interest. For this reason, practitioners often perform multiple assays based on different, overlapping combinations of markers. In this paper, we address the challenge of imputing the high dimensional jointly distributed values of marker attributes based on overlapping marginal observations. We show that simple nearest neighbor based imputation can lead to spurious subpopulations in the imputed data and introduce an alternative approach based on nearest neighbor imputation restricted to a cell's subpopulation. This requires us to perform clustering with missing data, which we address with a mixture model approach and novel EM algorithm. Since mixture model fitting may be ill-posed in this context, we also develop techniques to initialize the EM algorithm using domain knowledge. We demonstrate our approach on real flow cytometry data.

Keywords: Statistical file matching, Flow cytometry, Mixture model, Probabilistic PCA, EM algorithm, Imputation

*Corresponding author. Tel: +1 734 763 5228; Fax: +1 734 763 8041
Email addresses: gyemin@eecs.umich.edu (Gyemin Lee), wgfinn@umich.edu (William Finn), cscott@eecs.umich.edu (Clayton Scott)

[1] G. Lee and C. Scott were supported in part by NSF Award No. 0953135.
[2] G. Lee was supported in part by the Edwin R. Riethmiller Fellowship.

1. Introduction

Flow cytometry is a technique for quantitative cell analysis [1]. It provides simultaneous measurements of multiple characteristics of individual cells. Typically, a large number of cells are analyzed in a short period of time – up to thousands of cells per second. Since its development in the late 1960s, flow cytometry has become an essential tool in various biological and medical laboratories. Major applications of flow cytometry include hematological immunophenotyping and diagnosis of diseases such as acute leukemias, chronic lymphoproliferative disorders and malignant lymphomas [2].

Flow cytometry data has traditionally been analyzed by visual inspection of one-dimensional histograms or two-dimensional scatter plots. Clinicians will visually inspect a sequence of scatter plots based on different pairwise marker combinations and perform gating, the manual selection of marker thresholds, to eliminate certain subpopulations of cells. They identify various pathologies based on the shape of cell subpopulations in these scatter plots. In addition to traditional inspection-based analysis, there has been recent work, reviewed below, on automatic cell gating or classification of pathologies based on multidimensional analysis of cytometry data.

Unfortunately, flow cytometry analysis is limited by the number of markers that can be simultaneously measured. In clinical settings, this number is typically five to seven, while the number of markers of interest may be much larger. To overcome this limitation, it is common in practice to perform multiple assays based on different and overlapping combinations of markers. However, many marker combinations are never observed, which complicates scatter plot-based analysis, especially in retrospective studies. In addition, automated multidimensional analysis is not feasible because all cell measurements have missing values.

To address these issues, we present a statistical method for file matching, which imputes higher dimensional flow cytometry data from multiple lower dimensional data files. While Pedreira et al. [3] proposed a simple approach based on Nearest Neighbor (NN) imputation, this method is prone to induce spurious clusters, as we demonstrate below. Our method can improve the file matching of flow cytometry and is less likely to generate false clusters. The result is a full dataset, where arbitrary pairs can be viewed together, and multidimensional methods can be applied.

In the following, we explain the principles of flow cytometry and intro-

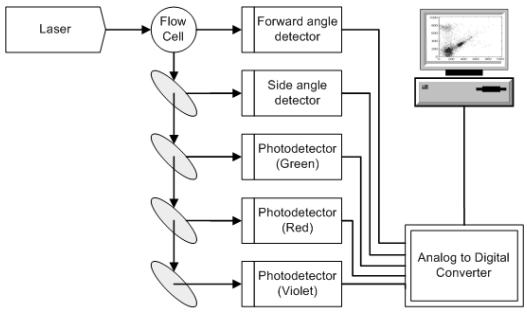

Figure 1: A flow cytometer system. As a stream of cells passes through a laser beam, photo-detectors detect forward angle light scatter, side angle light scatter and light emissions from fluorochromes. Then the digitized signals are analyzed in a computer.

duce the file matching problem in the context of flow cytometry data. We then present a file matching approach which imputes a cell's missing marker values with the values of the nearest neighbor among cells of the same type. To implement this approach, we develop a method for clustering with missing data. We model flow cytometry data with a latent variable Gaussian mixture model, where each Gaussian component corresponds to a cell type, and develop an expectation-maximization (EM) algorithm to fit the model. We also describe ways to incorporate domain knowledge into the initialization of the EM algorithm. We compare our method with nearest neighbor imputation on real flow cytometry data and show that our method offers improved performance. Our MATLAB implementation is available online at `http://www.eecs.umich.edu/~cscott/code/cluster_nn.zip`.

2. Background

In this section, we explain the principles of flow cytometry. We also define the statistical file matching problem in the context of flow cytometry data and motivate the need for an improved solution.

2.1. Flow cytometry

In flow cytometry analysis for hematological immunophenotyping, a cell suspension is first prepared from peripheral blood, bone marrow or lymph node. The suspension of cells is then mixed with a solution of fluorochrome-labeled antibodies. Typically, each antibody is labeled with a different fluorochrome. As the stream of suspended cells passes through a focused laser

beam, they either scatter or absorb the light. If the labeled antibodies are attached to proteins of a cell, the associated fluorochromes absorb the laser and emit light with a corresponding wavelength (color). Then a set of photodetectors in the line of the light and perpendicular to the light capture the scattered and emitted light. The signals from the detectors are digitized and stored in a computer system. Forward scatter (FS) and side scatter (SS) signals as well as the various fluorescence signals are collected for each cell (see Fig. 1).

For example, in a flow cytometer capable of measuring five attributes, the measurements of each cell can be represented with a 5-dimensional vector $\mathbf{x} = (x^{(1)}, \cdots, x^{(5)})$ where $x^{(1)}$ is FS, $x^{(2)}$ is SS and $x^{(3)}, \cdots, x^{(5)}$ are the fluorescent markers. We use "marker" to refer to both the biological entities and the corresponding measured attributes. Then the measurements of N cells are represented by vectors $\mathbf{x}_1, \cdots, \mathbf{x}_N$ and form a $N \times 5$ matrix.

The detected signals provide information about the physical and chemical properties of each cell analyzed. FS is related to the relative size of the cell, and SS is related to its internal granularity or complexity. The fluorescence signals reflect the abundance of expressed antigens on the cell surface. These various attributes are used for identification and quantification of cell subpopulations. FS and SS are always measured, while the marker combination is a part of the experimental design.

Flow cytometry data is usually analyzed using a sequence of one dimensional histograms and two or three dimensional scatter plots by choosing a subset of one, two or three markers. The analysis typically involves manually selecting and excluding cell subpopulations, a process called "gating", by thresholding and drawing boundaries on the scatter plots. Clinicians routinely diagnose by visualizing the scatter plots.

Recently, some attempts have been made to analyze directly in high dimensional spaces by mathematically modeling flow cytometry data. In [4, 5], a mixture of Gaussian distributions is used to model cell populations, while a mixture of t-distributions with a Box-Cox transformation is used in [6]. A mixture of skew t-distributions is studied in [7]. The knowledge of experts is sometimes incorporated as prior information [8]. Instead of using finite mixture models, some recent approaches proposed information preserving dimension reduction to analyze high dimensional flow cytometry data [9, 10]. However, standard techniques for multi-dimensional flow cytometry analysis are not yet established.

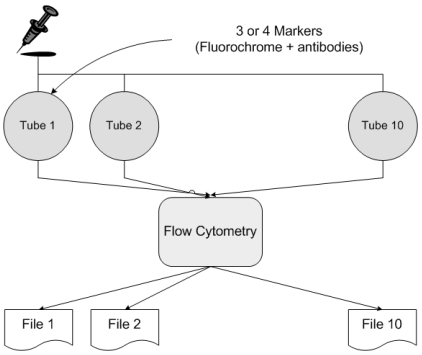

Figure 2: Flow cytometry analysis on a large number of antibody reagents within a limited capacity of a flow cytometer. A sample from a patient is separated into multiple tubes with which different combinations of fluorochrome-labeled antibodies are stained. Each output file contains at least two variables, FS and SS, in common as well as some variables that are specific to the file.

2.2. Statistical file matching

The number of markers used for selection and analysis of cells is constrained by the number of measurable fluorochrome channels (colors) in a given cytometer, which in turn is a function of the optical physics of the laser light source(s) and the excitation and emission spectra of the individual fluorochromes used to label antibodies to targeted surface marker antigens. Recent innovations have enabled measuring near 20 cellular attributes, through the use of multiple lasers of varying energy, multiple fluorochrome combinations, and complex color compensation algorithms. However, instruments deployed in clinical laboratories still only measure 5-7 attributes simultaneously [11].

There may be times in which it would be useful to characterize cell populations using more colors than can be simultaneously measured on a given cytometry platform. For example, some lymph node biopsy samples may be involved partially by lymphoma, in a background of hyperplasia of lymphoid follicles within the lymph node. In such cases, it can be useful to exclude the physiologic follicular lymphocyte subset based on a known array of marker patterns (for example, CD10 expression, brighter CD20 expression than non-germinal center B-cells, and CD38 expression) and evaluate the non-follicular lymphocyte fraction for markers known to be useful in the diagnosis of non-Hodgkin lymphomas (for example, CD5, CD19, CD23, kappa immunoglobulin light chain, and lambda immunoglobulin light chain). Un-

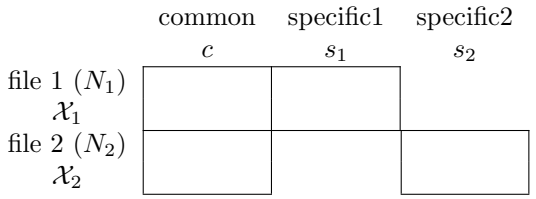

Figure 3: Data structure of two incomplete data files. The two files have some overlapping variables c and some variables s_1 and s_2 that are never jointly observed. File matching combines the two files by completing the missing blocks of variables.

less an 8-color (10 channel) flow cytometer is available, this analysis cannot be done seamlessly. In such case, the markers must be inferred indirectly, potentially resulting in dilution of the neoplastic lymphoma clone by normal background lymphocytes. Likewise, recent approaches to the analysis of flow cytometry data are built around the treatment of datasets as individual high-dimensional distributions or shapes, again limited only by the number of colors available in a given flow cytometry platform. Given the considerable expense of acquiring cytometry platforms capable of deriving high-dimensionality datasets, the ability to virtually combine multiple lower-dimensional datasets into a single high-dimensional dataset could provide considerable advantage in these situations.

When it is not possible to simultaneously measure all markers of interest, it is common to divide a sample into several "tubes" and stain each tube separately with a different set of markers (see Fig. 2) [12]. For example, consider an experiment with two tubes: Tube 1 containing 5000 cells is stained with CD45, CD5 and CD7, and Tube 2 containing 7000 cells is stained with CD45, CD10 and CD19. File 1 and File 2 record the FS, SS and marker measurements in the format of 5000×5 and 7000×5 matrices.

In the sequel, we present a method that combines two or more tubes and generates flow cytometry data in which all the markers of interest are available for the union of cells. Thus, we obtain a single higher dimensional dataset beyond the current limits of the instrumentation. Then pairs of markers that are not measured together can still be visualized through scatter plots, and methods of multidimensional analysis may potentially be applied to the full dataset.

This technique, called file matching, merges two or more datasets that have some commonly observed variables as well as some variables unique to each dataset. We introduce some notations to generalize the above example.

In Fig. 3, each unit (cell) $\mathbf{x}_n$ is a row vector in $\mathbb{R}^d$ and belongs to one of the data files (tubes) $\mathcal{X}_1$ or $\mathcal{X}_2$, where each file contains N_1 and N_2 units, respectively. While variables c are commonly observed for all units, variables s_2 are missing in $\mathcal{X}_1$ and s_1 are missing in $\mathcal{X}_2$, where s_1, s_2 and c indicate specific and common variable sets. If we denote the observed and missing components of a unit $\mathbf{x}_n$ with o_n and m_n, then $o_n = c \cup s_1$ and $m_n = s_2$ for $\mathbf{x}_n \in \mathcal{X}_1$ and $o_n = c \cup s_2$ and $m_n = s_1$ for $\mathbf{x}_n \in \mathcal{X}_2$.

Continuing the previous example, suppose that the attribute measurements are arranged as in Fig. 3 in the order of FS, SS, CD45, CD5, CD7, CD10 and CD19. Then each individual cell is seen as a row vector in $\mathbb{R}^7$ with two missing variables. Thus, $\mathcal{X}_1$ is a matrix with $N_1 = 5000$ rows and $\mathcal{X}_2$ is a matrix with $N_2 = 7000$ rows, and the common and specific attribute sets are $c = \{1, 2, 3\}$, $s_1 = \{4, 5\}$ and $s_2 = \{6, 7\}$.

A file matching algorithm impute the blocks of missing variables. Among imputation methods, conditional mean or regression imputations are most common. As shown in Fig. 4, however, these imputation algorithms tend to shrink the variance of the data. Thus, these approaches are inappropriate in flow cytometry where the shape of cell subpopulations is important in clinical analysis. More discussions on missing data analysis and file matching can be found in [13] and [14].

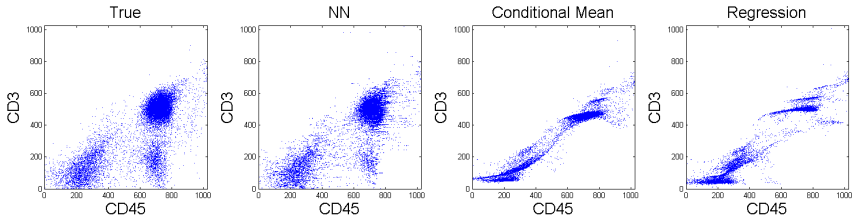

Figure 4: Examples of imputation methods: NN, conditional mean and regression. The NN method relatively well preserves the distribution of imputed data, while other imputation methods such as conditional mean and regression significantly reduce the variability of data.

A recent file matching technique in flow cytometry was developed by Pedreira et al. [3]. They proposed to use Nearest Neighbor (NN) imputation to match flow cytometry data files. In their approach, the missing variables of a unit, called the recipient, are imputed with the observed variables from a unit in the other file, called the donor, that is most similar. If $\mathbf{x}_i$ is a unit

in $\mathcal{X}_1$, the missing variables of $\mathbf{x}_i$ are set as follows:

$$\mathbf{x}_i^{m_i} = \mathbf{x}_j^{*\,m_i} \text{ where } \mathbf{x}_j^* = \underset{\mathbf{x}_j \in \mathcal{X}_2}{\arg\min} \|\mathbf{x}_i^c - \mathbf{x}_j^c\|_2.$$

Here $\mathbf{x}_i^{m_i} = (x_i^{(p)}, p \in m_i)$ and $\mathbf{x}_i^c = (x_i^{(p)}, p \in c)$ denote the row vectors of missing and common variables of $\mathbf{x}_i$, respectively. Note that the similarity is measured by the distance in the projected space of jointly observed variables. This algorithm is advantageous over other imputation algorithms using conditional mean or regression, as displayed in Fig. 4. It generally preserves the distribution of cells, while the other methods cause the variance structure to shrink toward zero.

However, the NN method sometimes introduces spurious clusters into the imputation results and fails to replicate the true distribution of cell populations. Fig. 5 shows an example of false clusters from the NN imputation algorithm (for the detailed experiment setup, see Section 4). We present a toy example to illustrate how NN imputation can fail and motivate our approach.

2.3. Motivating toy example

Fig. 6 shows a dataset in $\mathbb{R}^3$. In each file, only two of the three features are observed: c and s_1 in file 1 and c and s_2 in file 2. Each data point belongs to one of two clusters, but its label is unavailable. This example is not intended to simulate flow cytometry data, but rather to illustrate one way in which NN imputation can fail, and how our approach can overcome this limitation.

When imputing feature s_1 of units in file 2, the NN algorithm produces four clusters whereas there should be two, as shown in Fig. 6 (d). This is because the NN method uses only one feature and fails to leverage the information about the joint distribution of variables that are not observed together. However, if we can infer the cluster membership of data points, the NN imputation can be applied within the same cluster. Hence, we seek a donor from subgroup (1) for the data points in (3) and likewise we seek a donor from (2) for the points in (4) in the example. Then the file matching result greatly improves and better replicates the true distribution as in Fig. 6 (e).

In this example, as in real flow cytometry data, there is no way to infer cluster membership from the data alone, and incorrect labeling can lead to

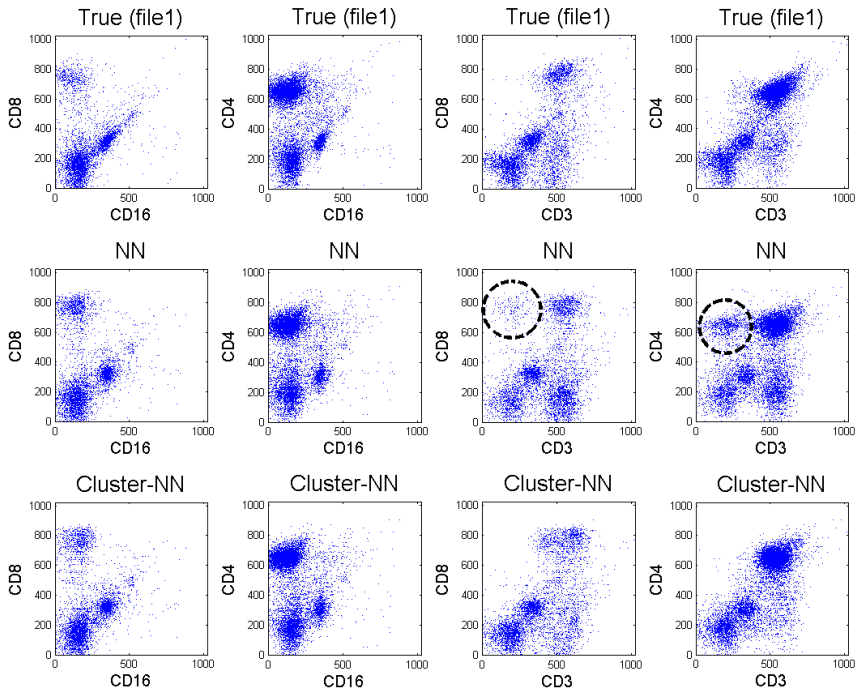

Figure 5: Comparison of results for two imputation methods to the ground truth cell distribution. Figures show scatter plots on pairs of markers that are not jointly observed. The middle row and the bottom row show the imputation results from the NN and the proposed Cluster-NN method, respectively. The results from the NN method show spurious clusters in the right two panels. The false clusters are indicated by dotted circles in the CD3 vs. CD8 and CD3 vs. CD4 scatter plots. On the other hand, the results from our proposed approach better resemble the true distribution on the top row.

poor results (Fig. 6 (f)). Fortunately, in flow cytometry we can incorporate domain knowledge to achieve an accurate clustering.

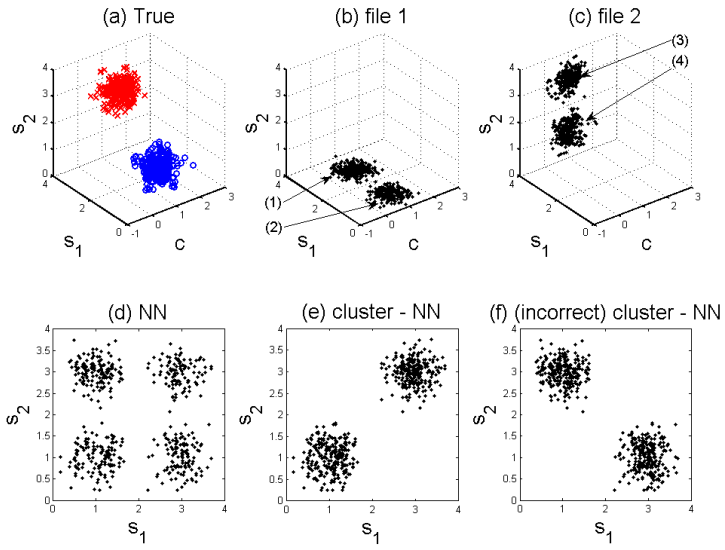

Figure 6: Toy example of file matching. Two files (b) and (c) provide partial information of data points (a) in $\mathbb{R}^3$. The variable c is observed in both files while s_1 and s_2 are specific to each file. The NN method created false clusters in the s_1 vs. s_2 scatter plot in (d). On the other hand, the proposed Cluster-NN method, which applies NN within the same cluster, successfully replicated the true distribution. If the clusters are incorrectly paired, however, the Cluster-NN approach can fail, as in (f).

3. Methods

3.1. Cluster-based imputation of missing variables

We first focus on the case of matching two files. The case of more than two files is discussed in Section 5. For the present section, we assume that there is a single underlying distribution with K clusters, and each $\mathbf{x} \in \mathcal{X}_1$ and each $\mathbf{x} \in \mathcal{X}_2$ is assigned to one of these clusters. Let $\mathcal{X}_1^k$ and $\mathcal{X}_2^k$ denote the cells in $\mathcal{X}_1$ and $\mathcal{X}_2$ from the kth cluster, respectively.

Suppose that the data is configured as in Fig. 3. In order to impute the missing variables of a recipient unit in $\mathcal{X}_1$, we locate a donor among the data points in $\mathcal{X}_2$ that have the same cluster label as the recipient. When

Input: Two files $\mathcal{X}_1$ and $\mathcal{X}_2$ to be matched
1. Jointly cluster the units in $\mathcal{X}_1$ and $\mathcal{X}_2$.
2. Perform Nearest Neighbor imputation within the same cluster.

Output: Statistically matched complete files $\widehat{\mathcal{X}}_1$ and $\widehat{\mathcal{X}}_2$

Figure 7: The description of the proposed Cluster-NN algorithm for two files.

imputing incomplete units in $\mathcal{X}_2$, the roles change. The similarity between two units is evaluated on the projected space of jointly observed variables, while constraining both units to belong to the same cluster. Then we impute the missing variables of the recipient by patching the corresponding variables from the donor. More specifically, for $\mathbf{x}_i \in \mathcal{X}_1^k$, we impute the missing variables by

$$\mathbf{x}_i^{m_i} = \mathbf{x}_j^{*m_i} \text{ where } \mathbf{x}_j^* = \underset{\mathbf{x}_j \in \mathcal{X}_2^k}{\arg\min} \|\mathbf{x}_i^c - \mathbf{x}_j^c\|_2.$$

Fig. 7 describes the proposed Cluster-NN imputation algorithm.

In social applications such as survey completion, file matching is often performed on the same class such as gender, age, or county of residence [14]. Unlike our algorithm, however, the information for labeling each unit is available in those applications and the class inference step is unnecessary.

3.2. Clustering with missing data

To implement the above approach, it is necessary to cluster the flow cytometry data. Thus, we concatenate two input files $\mathcal{X}_1$ and $\mathcal{X}_2$ into a single dataset as in Fig. 3. We model the data with a mixture model with each component of the mixture corresponding to a cluster. We emphasize that we are jointly clustering $\mathcal{X}_1$ and $\mathcal{X}_2$, not each file separately. Thus, each $\mathbf{x}$ in the merged dataset is assigned to one of the K mixture model components.

In a mixture model framework, the probability density function of a d-dimensional data vector $\mathbf{x}$ takes the form

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \, p_k(\mathbf{x})$$

where π_k are mixing weights of K components and p_k are component density functions. In flow cytometry, mixture models are widely-used to model cell

populations. Among mixture models, Gaussian mixture models are common [4, 5, 8], while distributions with more parameters, such as t-distributions, skew normal or skew t-distributions, have been recently proposed [6, 7]. While non-Gaussian models might provide a better fit, there is a trade-off between bias and variance. More complicated models tend to be more challenging to fit. Furthermore, even with an imperfect data model, we may still achieve an improved file matching.

Clustering amounts to fitting the parameters of the mixture model to the data points in $\mathcal{X}_1$ and $\mathcal{X}_2$. Given the model, a data point $\mathbf{x}$ is assigned to cluster k for which the posterior probability is maximized. Here we explain the mixture model that we used to model the cell populations (Section 3.2.1) and present an EM algorithm for inferring the model parameters, which determine the cluster membership of each data point (Section 3.2.2).

3.2.1. Mixture of PPCA

Fitting multidimensional mixture models require estimating a large number of parameters, and obtaining reliable estimates becomes difficult when the number of components or the dimension of the data increase. Here we adopt a probabilistic principal component analysis (PPCA) mixture model as a way to concisely model cell populations.

PPCA was proposed by [15] as a probabilistic interpretation of PCA. While conventional PCA lacks a probabilistic formulation, PPCA specifies a generative model, in which a data vector is linearly related to a latent variable. The latent variable space is generally lower dimensional than the ambient variable space, so the latent variable provides an economical representation of the data. Our motivations for using PPCA over a full Gaussian mixture model are that the parameters can be fit more efficiently (as demonstrated in Section 4), and in higher dimensional settings, a full Gaussian mixture model may have too many parameters to be accurately fit.

The PPCA model is built by specifying a distribution of a data vector $\mathbf{x} \in \mathbb{R}^d$ conditional on a latent variable $\mathbf{t} \in \mathbb{R}^q$, $p(\mathbf{x}|\mathbf{t}) = \mathcal{N}(\mathbf{W}\mathbf{t} + \boldsymbol{\mu}, \sigma^2\mathbf{I})$ where $\boldsymbol{\mu}$ is a d-dimensional vector and $\mathbf{W}$ is a $d \times q$ linear transform matrix. Assuming the latent variable $\mathbf{t}$ is normally-distributed, $p(\mathbf{t}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, the marginal distribution of $\mathbf{x}$ becomes Gaussian $p(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}, \mathbf{C})$ with mean $\boldsymbol{\mu}$ and covariance matrix $\mathbf{C} = \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}$. Then the posterior distribution can be shown to be Gaussian as well: $p(\mathbf{t}|\mathbf{x}) = \mathcal{N}(\mathbf{M}^{-1}\mathbf{W}^T(\mathbf{x} - \boldsymbol{\mu}), \sigma^2\mathbf{M}^{-1})$ where $\mathbf{M} = \mathbf{W}^T\mathbf{W} + \sigma^2\mathbf{I}$ is a $q \times q$ matrix.

The PPCA mixture model is a combination of multiple PPCA compo-

nents. This model offers a way of controlling the number of parameters to be estimated without completely sacrificing the model flexibility. In the full Gaussian mixture model, each Gaussian component has $d(d+1)/2$ covariance parameters if a full covariance matrix is used. The number of parameters can be reduced by constraining the covariance matrix to be isotropic or diagonal. However, these are too restrictive for cell populations since the correlation structure between variables cannot be captured. On the other hand, the PPCA mixture model lies between those two extremes and allows control of the number of parameters through specification of q, the dimension of the latent variable.

A PPCA mixture can be viewed as a Gaussian mixture with structured covariances. In Gaussian mixtures, various approaches constraining covariance structures have been proposed [16], where each cluster is required to share parameters to have the same orientation, volume or shape. However, in the PPCA model, the geometry of each cluster is allowed to vary between clusters, and the cluster parameters for different clusters are not constrained to be related to one another. Therefore, the PPCA mixture model is preferable in flow cytometry where cell populations typically have different geometric characteristics.

In a mixture of PPCA model, each PPCA component explains local data structure or a cell subpopulation, and the collection of component parameters $\theta_k = \{\pi_k, \boldsymbol{\mu}_k, \mathbf{W}_k, \sigma_k^2\}$, $k = 1, \cdots, K$, defines the model. An EM algorithm can learn the model by iteratively estimating these parameters. More details on the PPCA mixture and the EM algorithm for data without missing values are explained in [17].

3.2.2. Missing data EM algorithm

The concatenated dataset of $\mathcal{X}_1$ and $\mathcal{X}_2$ contains only partial observations of $N = N_1 + N_2$ units. Hence, we cannot directly apply the EM algorithm for a PPCA mixture to infer the model parameters. In the present section, we devise a novel EM algorithm for the missing data.

Even though our file matching problem has a particular pattern of missing variables, we develop a more general algorithm that allows for an arbitrary pattern of missing variables. Our development assumes values are "missing at random," meaning that whether a variable is missing or not is independent of its value [13]. We note that [18] presented an EM algorithm for a Gaussian mixture with missing data, and [17] presented EM algorithms for a PPCA mixture when data is completely observed. Therefore, our algorithm may be

viewed as an extension of the algorithm of [18] to PPCA mixtures, or the algorithm of [17] to data with missing values.

Denoting the observed and missing variables by o_n and m_n, each data point can be divided as $\mathbf{x}_n = \begin{bmatrix} \mathbf{x}_n^{o_n} \\ \mathbf{x}_n^{m_n} \end{bmatrix}$. Recall that, in the file matching problem, o_n indexes the union of common variables and the observed specific variables, and m_n indexes the unobserved specific variables so that $x_n^{(i)}$, $i \in o_n$, are observed variables and $x_n^{(i)}$, $i \in m_n$, are missing variables. This is only for notational convenience and does not imply that the vector $\mathbf{x}_n$ is re-arranged to this form.

Thus, we are given a set of partial observations $\{\mathbf{x}_1^{o_1}, \cdots, \mathbf{x}_N^{o_N}\}$. To invoke the EM machinery, we introduce indicator variables $\mathbf{z}_n$. One and only one entry of $\mathbf{z}_n$ is nonzero and $z_{nk} = 1$ indicates that the kth component is responsible for generating $\mathbf{x}_n$. We also include the missing variables $\mathbf{x}_n^{m_n}$ and the set of latent variables $\mathbf{t}_{nk}$ for each component to form the complete data $(\mathbf{x}_n^{o_n}, \mathbf{x}_n^{m_n}, \mathbf{t}_{nk}, \mathbf{z}_n)$ for $n = 1, \cdots, N$ and $k = 1, \cdots, K$.

We derive an EM algorithm for the PPCA mixture model with missing data. The key difference from the EM algorithm for completely observed data is that the conditional expectation is taken with respect to $\mathbf{x}^o$ as opposed to $\mathbf{x}$ in the expectation step.

To develop an EM algorithm, we employ and extend the two-stage procedure as described in [17]. In the first stage of the algorithm, the component weights π_k and the component center $\boldsymbol{\mu}_k$ are updated:

$$\widehat{\pi}_k = \frac{1}{N} \sum_n \langle z_{nk} \rangle \,, \tag{1}$$

$$\widehat{\boldsymbol{\mu}}_k = \frac{\sum_n \langle z_{nk} \rangle \begin{bmatrix} \mathbf{x}_n^{o_n} \\ \langle \mathbf{x}_n^{m_n} \rangle \end{bmatrix}}{\sum_n \langle z_{nk} \rangle} \tag{2}$$

where $\langle z_{nk} \rangle = P(z_{nk} = 1 | \mathbf{x}_n^{o_n})$ is the responsibility of mixture component k for generating the unit $\mathbf{x}_n$ and $\langle \mathbf{x}_n^{m_n} \rangle = \mathbb{E}[\mathbf{x}_n^{m_n} | z_{nk} = 1, \mathbf{x}_n^{o_n}]$ is the conditional expectation. Note that we are not assuming the vectors in the square bracket are arranged to have this pattern. This notation can be replaced by the true variable ordering.

Cell Type	CD markers
granulocytes	CD45+, CD15+
monocytes	CD45+, CD14+
helper T cells	CD45+, CD3+
cytotoxic T cells	CD45+, CD3+, CD8+
B cells	CD45+, CD19+ or CD45+, CD20+
Natural Killer cells	CD16+, CD56+, CD3-

Table 1: Types of human white blood cells. The T cells, B cells and NK cells are called lymphocytes. Each cell type is characterized by a set of expressed cluster of differentiation (CD) markers. The CD markers are commonly used to identify cell surface molecules on white blood cells. The '+/−' signs indicate whether a certain cell type has the corresponding antigens on the cell surface.

In the second stage, we update $\mathbf{W}_k$ and σ_k^2:

$$\widehat{\mathbf{W}}_k = \mathbf{S}_k \mathbf{W}_k (\sigma_k^2 \mathbf{I} + \mathbf{M}_k^{-1} \mathbf{W}_k^T \mathbf{S}_k \mathbf{W}_k)^{-1}, \tag{3}$$

$$\widehat{\sigma}_k^2 = \frac{1}{d} \mathrm{tr}\left(\mathbf{S}_k - \mathbf{S}_k \mathbf{W}_k \mathbf{M}_k^{-1} \widehat{\mathbf{W}}_k^T\right) \tag{4}$$

from local covariance matrix $\mathbf{S}_k$:

$$\mathbf{S}_k = \frac{1}{N \widehat{\pi}_k} \sum_n \langle z_{nk} \rangle \left\langle \left(\begin{bmatrix} \mathbf{x}_n^{o_n} \\ \langle \mathbf{x}_n^{m_n} \rangle \end{bmatrix} - \widehat{\boldsymbol{\mu}}_k \right) \left(\begin{bmatrix} \mathbf{x}_n^{o_n} \\ \langle \mathbf{x}_n^{m_n} \rangle \end{bmatrix} - \widehat{\boldsymbol{\mu}}_k \right)^T \right\rangle.$$

The new parameters are denoted by $\widehat{\pi}_k, \widehat{\boldsymbol{\mu}}_k, \widehat{\mathbf{W}}_k$ and $\widehat{\sigma}_k^2$. These update rules boil down to the update rules for completely observed data when there are no missing variables. We derive the EM algorithm in detail in Appendix A.

After model parameters are estimated, the observations are divided into groups according to their posterior probabilities:

$$\arg\max_{k=1,\cdots K} p\left(z_{nk} = 1 | \mathbf{x}_n^{o_n}\right),$$

so each unit (cell) is classified into one of K cell subpopulations. Note that this posterior probability is computed in the E-step. This gives the desired clustering.

3.2.3. Domain knowledge and initialization of EM algorithm

Because of the missing data, fitting a PPCA mixture model is ill-posed, in the sense that several local maxima of the likelihood may explain the data

equally well. For example, in the toy example in Section 2.3, there is no way to know the correct cluster inference based solely on the data. However, we can leverage domain knowledge to select the number of components and initialize model parameters.

In flow cytometry, from the design of fluorochrome marker combinations and knowledge about the blood sample composition, we can anticipate certain properties of cell subpopulations. For example, Table 1 summarizes white blood cell types and their characteristic cluster of differentiation (CD) marker expressions. The six cell types suggests choosing $K = 6$ when analyzing white blood cells.

The CD markers indicated are commonly used in flow cytometry to identify cell surface molecules on leukocytes (white blood cells) [19]. However, this information is qualitative and needs to be quantified. Furthermore, the appropriate quantification depends on the patient and flow cytometry system.

To achieve this, we use one dimensional histograms. In a histogram, two large peaks are generally expected depending on the expression level of the corresponding CD marker. If a cell subpopulation expresses a CD marker, denoted by '+', then it forms a peak on the right side of the histogram. On the other hand, if a cell subpopulation does not express the marker, denoted by '−', then a peak can be found on the left side of the histogram. We use the locations of the peaks to quantify the expression levels.

These quantified values can be combined with the CD marker expression levels of each cell type to specify the initial cluster centers. Thus, each element of $\boldsymbol{\mu}_k$ of a certain cell type is initialized by either the positive quantity or the negative quantity from the histogram. In our implementation, these are set manually by visually inspecting the histograms. Then we initialize the mixture model parameters $\{\pi_k, \boldsymbol{\mu}_k, \mathbf{W}_k, \sigma_k^2\}$ as described in Fig. 8.

An important issue in file matching arises from the covariance matrix. When data is completely observed, a common way of initializing a covariance matrix is using a sample covariance matrix. In the case of file matching, however, it cannot be evaluated since some sets of variables are never jointly observed (see Fig. 9). Hence, we build $\mathbf{C}_k$ from variable to variable with sample covariances, whenever possible. For example, we can set $\mathbf{C}_k^{c,s_1}$ with the sample covariance of data points in $\mathcal{X}_1$ where variables c and s_1 are available. On the other hand, the submatrix $\mathbf{C}_k^{s_1,s_2}$ cannot be built from observations. In our implementation, we set the submatrix $\mathbf{C}_k^{s_1,s_2}$ randomly from a standard normal distribution. However, the resulting matrix may

Input: $\mathcal{X}_1$, $\mathcal{X}_2$ data files; K the number of components; q the dimension of the latent variable space; $\boldsymbol{\mu}_k$ the initial component means.

for $k = 1$ to K **do**

1. Using distance $\|\mathbf{x}_n^{o_n} - \boldsymbol{\mu}_k^{o_n}\|_2$, find the set of data points $\mathcal{X}^k$ whose nearest component mean is $\boldsymbol{\mu}_k$
2. Initialize observable submatrices of $\mathbf{C}_k$ with sample covariances of data in $\mathcal{X}^k$, and the remaining entries with random draws from a standard normal distribution.
3. Make $\mathbf{C}_k$ positive definite by replacing negative eigenvalues with a tenth of the smallest positive eigenvalue.
4. Set $\pi_k = |\mathcal{X}^k|/(N_1 + N_2)$
5. Set $\mathbf{W}_k$ with the q principal eigenvectors of $\mathbf{C}_k$
6. Set σ_k^2 with the average of remaining eigenvalues of $\mathbf{C}_k$

end for

Output: $\{\pi_k, \boldsymbol{\mu}_k, \mathbf{W}_k, \sigma_k^2\}$ for $k = 1, \cdots, K$

Figure 8: Parameter initialization of an EM algorithm for missing data. Cell populations are partitioned into K groups by the distance to each component center. The component weight π_k is proportional to the size of each partition. From the covariance matrix estimate $\mathbf{C}_k$, parameters $\mathbf{W}_k$ and σ_k^2 are initialized by taking the eigen-decomposition.

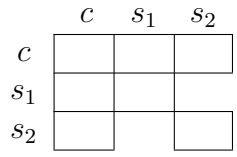

Figure 9: Structure of covariance matrix $\mathbf{C}$. The sub-matrices $\mathbf{C}_k^{s_1,s_2}$ and $\mathbf{C}_k^{s_2,s_1}$ cannot be estimated from a sample covariance matrix because these variables are never jointly observed.

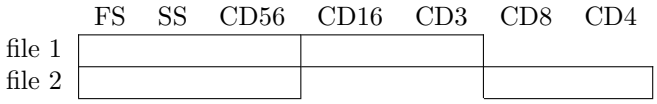

Figure 10: File structure used in the single tube experiments in Section 4.1. FS, SS and CD56 are common in both files, and a pair of CD markers are observed in only one of the files. The blank blocks correspond to the unobserved variables. The blocks in file 1 are matrices with N_1 rows and the blocks in file 2 are matrices with N_2 rows.

not be positive definite. Thus, we made $\mathbf{C}_k$ positive definite by replacing negative eigenvalues with a tenth of the smallest positive eigenvalue. Once a covariance matrix $\mathbf{C}_k$ is obtained, we can initialize $\mathbf{W}_k$ and σ_k^2 by taking the eigen-decomposition of $\mathbf{C}_k$.

4. Results

We apply the proposed file matching technique to real flow cytometry datasets and present experimental results. Three flow cytometry datasets were prepared from lymph node samples of three patients. These datasets were provided by the Department of Pathology at the University of Michigan.

We consider two experimental settings. In the first experiment (Section 4.1), we artificially create two incomplete data files from a single tube and compare the imputed results to the original true dataset. In the second experiments (Section 4.2), we investigate multiple tubes where each file is derived individually from two different tubes and the imputed results are compared to separate reference data.

4.1. Single tube experiments

From each patient sample, a dataset is obtained with seven attributes: FS, SS, CD56, CD16, CD3, CD8 and CD4. Two files are built from this dataset, and two attributes from each file are made hidden to construct hypothetical missing data. Hence, CD16 and CD3 are available only in file 1, and CD8 and CD4 are available only in file 2, while FS, SS and CD56 are commonly available in both files. Fig. 10 illustrates the resulting data pattern where the blocks of missing variables are left blank.

For each white blood cell type, its expected marker expressions (CD markers), relative size (FS) and relative granularity (SS) are presented in Table 2. The '+/−' signs indicate whether a certain type of cells expresses the markers or not. For example, helper T cells express both CD3 and CD4 but not

Cell type	FS	SS	CD56	CD16	CD3	CD8	CD4
granulocytes	+	+	−	+	−	−	−
monocytes	+	−	−	+	−	−	−
helper T cells	−	−	−	−	+	−	+
cytotoxic T cells	−	−	−	−	+	+	−
B cells	−	−	−	−	−	−	−
Natural Killer cells	−	−	+	+	−	−	−

Table 2: Cell types and their corresponding marker expressions for data in the single tube experiments. '+' or '−' indicates whether a certain cell type expresses the CD marker or not.

others. As explained in Section 3.2.3, we quantify this qualitative knowledge with the help of one dimensional histograms. Two dominant peaks corresponding to the positive and negative expression levels are picked from each histogram, and their measurement values are set to the expression levels. Fig. 11 and Table 3 summarize this histogram analysis. When two negative peaks are present as in CD8, the stronger ones are chosen in our implementation. In flow cytometry, it is known that two types of cells with the same '−' marker can cause slightly different measurement levels. However, this difference between '−' peaks is often small and less significant compared to the difference between '+' and '−' peaks. When we tried experiments (not presented) by choosing weaker peaks, we could not observe meaningful changes in the results.

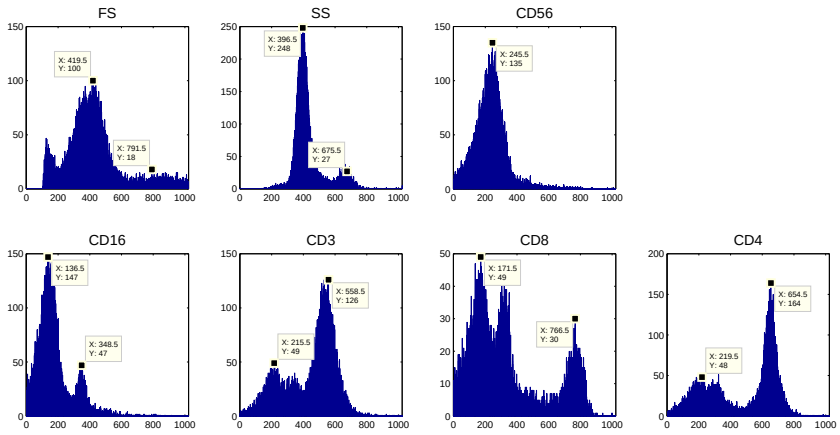

Figure 11: Histogram of each marker in the single tube experiments (Section 4.1). The peaks are selected manually and are indicated in each panel.

	FS	SS	CD56	CD16	CD3	CD8	CD4
+	800	680	500	350	550	750	650
−	400	400	240	130	200	170	200

Table 3: The positive and negative expression levels are extracted from the histograms in Fig. 11. These values are used to initialize the EM algorithm.

Following the procedure delineated in Section 3, two incomplete data files are completed. A mixture of PPCA is fitted with six components because six cell types are expected on this dataset. The latent variable dimension of each PPCA component is fixed to two. The convergence of the missing data EM algorithm is determined when the relative change of log-likelihood value is less than 10^{-10} or the number of iterations reaches 5000. Fig. 12 shows the evolution as iteration continues. The likelihood value increases sharply during the dozens of steps in the beginning and then converges.

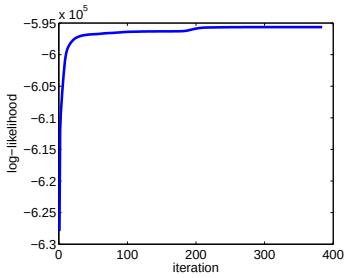

Figure 12: Typical convergence of the proposed missing data EM algorithm.

The synthesized data after file matching is displayed in Fig. 5. The figure shows scatter plots of specific variables: CD16, CD3, CD4 and CD8. Note that these marker pairs are not available from any of the two incomplete data files, while other marker pairs are directly obtainable from the observed cells. The imputation results from the NN and the Cluster-NN methods are compared in the figure. For reference, the figure also presents scatter plots of the ground truth dataset. As can be seen, the results from the Cluster-NN better coincide with the true distributions. By contrast, the NN method generates spurious clusters in the CD3-CD8 and CD3-CD4 scatter plots, and the results are far from the true distributions. These false clusters are indicated in Fig. 5. We quantify the quality of the imputed values below in Section 4.3.

Figure 13: Data pattern used in the multiple tube experiments in Section 4.2. Both files contain FS, SS and CD5 commonly, and each file contains one of CD45 and CD19. All marker attributes are available in a separate reference file.

4.2. Multiple tube experiments

In this second experiment, we involve multiple tubes and demonstrate the file matching of flow cytometry data.

Two tubes from each of the three patient samples are stained individually with different marker combinations: CD5/CD45 and CD5/CD19. For comparison with actually measured data, an additional tube is conjugated with markers CD5/CD45/CD19. This additional tube dataset is used only for evaluation of imputation results and is not involved during the file matching. Fig. 13 illustrate the pattern of datasets used in the experiments.

As opposed to the previous single tube experiments, the experiments on multiple tubes impose another complication. It is well-known that in flow cytometry, the instrument can drift over time. This technical variation causes the shifts in population positions. To minimize the effects from this variation, data files can be preprocessed with normalization techniques [20, 21] before applying file matching algorithms.

However, the rate of this drift is typically very slow and on a much larger scale than the time for one set of tubes. Furthermore, operators are careful to calibrate each tube (based on the same sample) in the same way to minimize such variation. For these reasons, technical variation within a batch of tubes corresponding to the same patient/sample is much less of an issue in flow cytometry, compared to technical variation between data gathered at different times. Since no noteworthy population shift was found from the histogram analysis in Fig. 14, we proceeded without any normalization.

For datasets in multi-tube experiments, Table 4 shows the relative marker expression levels of various types of white blood cells. Their corresponding numerical measurement levels are found from this table and the histograms in Fig. 14, and given in Table 5. Since all white blood cells express CD45, its negative level is left blank in the table.

Similarly to the above experiments, the two incomplete data files are imputed using the Cluster-NN algorithm as explained in Section 3. In this experiment, a PPCA mixture model with five components is fitted to the

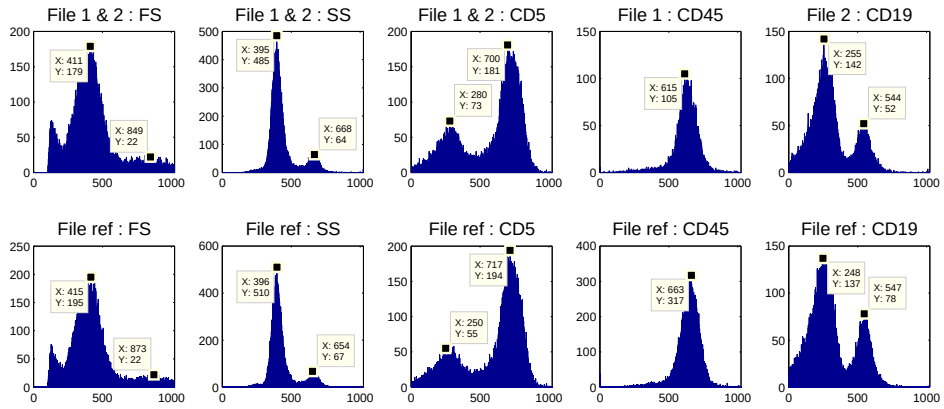

Figure 14: The top row shows histograms from the two incomplete files. Histograms from the reference file are shown in the bottom row. The peaks of each marker are indicated. No noticeable population shift across files was observable.

Cell type	FS	SS	CD5	CD45	CD19
granulocytes	+	+	−	+	−
monocytes	+	−	−	+	−
helper T cells	−	−	+	+	−
cytotoxic T cells	−	−	+	+	−
B cells	−	−	−	+	+
Natural Killer cells	−	−	−	+	−

Table 4: Types of white blood cells and their corresponding markers expressions for data in the multiple tube experiments..

	FS	SS	CD5	CD45	CD19
+	850	670	700	615	545
−	410	395	280	−	255

Table 5: The positive and negative expression levels are obtained from the histograms in Fig. 14. Since all white blood cells express CD45, the negative level is left blank.

missing data. We choose five components because the two types of T cells share the same row in Table 4. The dimension of the latent variable of each component, q, is set to two.

Fig. 15 displays the cell distributions of imputed data files. The presented marker pair CD45-CD19 is not originally available in any of the two files in experiments. The corresponding scatter plot from the separate reference file is also drawn. While the imputed results from the NN method and the Cluster-NN method look similar, a horizontal drift of cells in high CD19 subpopulation can be observed in the NN result. This spread of cells is not present in the reference plot and the Cluster-NN result.

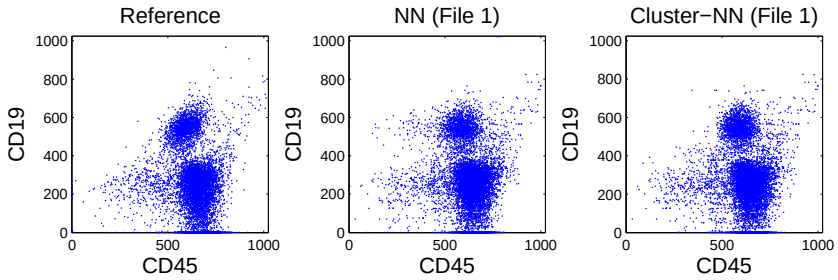

Figure 15: Comparison of two imputation results with the actual measurements in the reference file. The result from the NN method shows a horizontal drift of cells in high CD19 population. This is not observed in the Cluster-NN result and the reference file.

4.3. Evaluation method

To quantitatively evaluate the previous results, we use Kullback-Leibler (KL) divergence. The KL divergence between two distribution $f(\mathbf{x})$ and $g(\mathbf{x})$ is defined by

$$KL(g \,\|\, f) = \mathbb{E}_g \left[\log g - \log f \right].$$

Let f be a true distribution responsible for the observations and g be its estimate.

The KL divergence is asymmetric and $KL(f \,\|\, g)$ and $KL(g \,\|\, f)$ have different meanings. We prefer $KL(g \,\|\, f)$ to $KL(f \,\|\, g)$ because the former more heavily penalize the over-estimation of the support of f. This allows us to assess when an imputation method introduces spurious clusters.

For the single tube and the multiple tube experiments, we evaluated the KL divergence of the imputation results. We randomly permuted each

ID	NN (file 1)	Cluster-NN (file 1)	NN (file 2)	Cluster-NN (file 2)
Patient1	2.90 ± 0.05	1.55 ± 0.05	2.66 ± 0.03	1.12 ± 0.04
Patient2	4.54 ± 0.07	1.22 ± 0.03	4.12 ± 0.08	0.92 ± 0.03
Patient3	4.46 ± 0.10	2.40 ± 0.11	4.18 ± 0.11	2.30 ± 0.07

(a) Single tube experiments

ID	NN (file 1)	Cluster-NN (file 1)	NN (file 2)	Cluster-NN (file 2)
Patient1	0.51 ± 0.01	0.46 ± 0.02	0.41 ± 0.01	0.40 ± 0.01
Patient2	0.64 ± 0.01	0.62 ± 0.03	0.80 ± 0.03	0.78 ± 0.04
Patient3	0.88 ± 0.05	0.78 ± 0.07	0.80 ± 0.02	0.65 ± 0.03

(b) Multiple tube experiments

Table 6: The KL divergences are computed for ten permutations of each flow cytometry dataset. The averages and standard errors are reported in the table. For both the NN and Cluster-NN algorithm, the file matching results are evaluated. (a) In the single tube experiments, the KL divergences of Cluster-NN are closer to zero than those of NN. Thus, the results from Cluster-NN better replicated the true distribution. (b) In the multiple tube experiments, the Cluster-NN consistently performed better than the NN. However, the differences between two algorithms are small.

dataset ten times, and divided into incomplete data files and evaluation sets. Then we computed the KL divergence for each permutation, and reported their averages and standard errors in Table 6. The details of dividing datasets and computing the KL divergence are explained in Appendix B.

As can be seen, the KL divergences from Cluster-NN are substantially smaller than those from NN in the first set of experiments on a single tube. Therefore, the Cluster-NN yielded a better replication of true distribution. In the second series of experiments, the differences in KL divergence between algorithms were minor. While we could observe the spread of cells in the NN results (see Fig. 15), their effect on the KL divergence was sometimes small due to their relatively small number.

4.4. Computational considerations

Here we consider computational aspects of the PPCA mixture model and its EM algorithm.

As we described above in Section 3.2.1, through the PPCA mixtures, we can control the number of model parameters without losing the model flexibility. When combining more tubes, this ensures that there is sufficient data for parameter estimation with higher dimensionality. Another advantage of using PPCA mixtures is the execution time of the EM algorithm. Under Windows 7 system equipped with two Intel(R) Xeon(R) 2.27 GHz processors

and RAM 12GB, the average convergence time with PPCA mixtures was about 23 seconds in the above single tube experiment. On the contrary, it took nearly 200 seconds on average to fit full Gaussian mixtures. That is, fitting a PPCA mixture model took approximately eight times less relative to one based on a full Gaussian mixture model. This computational improvement is highly desirable because demands for high-throughput analysis are sharply increasing in flow cytometry.

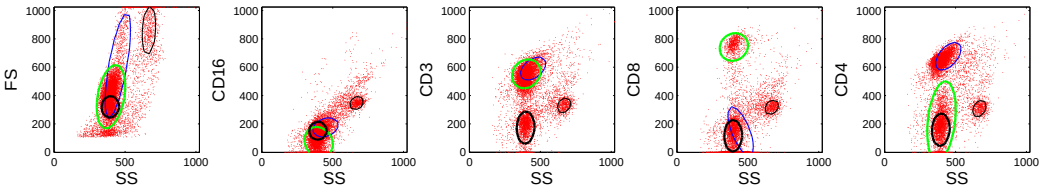

Figure 16: The scatter plots of a dataset used in the single tube experiments (Section 4.1) are drawn on several marker pairs. The fitted mixture components are shown as well on each panel. For clarity, four among the six components are displayed.

During the series of experiments, we have chosen the number of mixture components based on the number of cell types. Then mixture models are learned from the partially observed data. Fig. 16 illustrates how the clustering behaves on a dataset used in the single tube experiments (Section 4.1). Component contours are overlaid on the scatter plots over a few observed marker pairs. Most contours can be successfully identified with important cell subpopulations in the dataset, while there are some cases where we could not find the corresponding cell types.

Although many mixture model-based analysis in flow cytometry rely on criteria such as Akaike information criterion or Bayesian information criterion to select the number of components [5, 6, 7], these approaches assume completely observed data, whereas most of the data are missing in file matching. In practice, a good rule of thumb is to set the number of mixture components with the number of cell types. Fig. 17 shows the effect of the number of components. For a range of K, where K is the number of components, we repeated the single tube experiments in Section 4.1. Six points are first selected from from Fig. 11 and Table 2, and then used for $K = 6$. For models with more or less than 6 components, each centroid is initialized by random drawing from a Gaussian distribution centered at one of the six points. Once cluster centers are initialized, the rest of the parameters are initialized by following the method described in Fig. 8. The best performance is given

when $K = 7$, with the performance slightly better than the performance when $K = 6$. For values of K less than 6, the performance was much worse, and for values greater than 7, the performance gradually degraded as the number of components was increased.

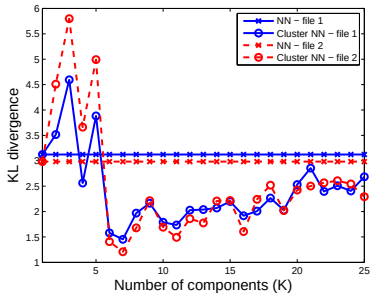

Figure 17: The KL divergence of Cluster-NN imputation results over the number of components of a PPCA mixture model. As the NN method does not involve clustering, the KL divergence remains constant. The best performance of Cluster-NN is achieved near $K = 7$.

5. Discussion

In this paper, we demonstrated the use of a cluster-based nearest neighbor (Cluster-NN) imputation method for file matching of flow cytometry data. We applied the proposed algorithm on real flow cytometry data to generate a dataset of higher dimension by merging two data files of lower dimensions. The resulting matched file can be used for visualization and high-dimensional analysis of cellular attributes.

While the presented imputation method focused on the case of two files, it can be generalized to more than two files. We envision two possible extensions of the Cluster-NN imputation method. For concreteness, suppose that five files $\mathcal{X}_1, \cdots, \mathcal{X}_5$ are given and a missing variable of $\mathbf{x}_i \in \mathcal{X}_1$ is available in $\mathcal{X}_2$ and $\mathcal{X}_3$.

Method 1. The first approach fits a single mixture of PPCA model to the all units in the five files using the missing data EM algorithm in Section 3.2. According to their posterior probabilities, units in each file are clustered into classes. If $\mathbf{x}_i$ belongs to $\mathcal{X}_1^k$, then the similarities are computed between $\mathbf{x}_i$ and units in $\mathcal{X}_2^k$ and $\mathcal{X}_3^k$. Then the most similar unit is chosen to be the donor.

Method 2. In the second method, a pair of files are considered at a time by selecting and limiting the search for a donor to one of $\mathcal{X}_2$ and $\mathcal{X}_3$. One can pick a file with more cells, say $\mathcal{X}_3$. Thus, the donor candidates are found among units in $\mathcal{X}_3$. Then the PPCA mixture model is trained with the cells in $\mathcal{X}_1$ and $\mathcal{X}_3$ using the missing data EM algorithm. After units in $\mathcal{X}_1$ and $\mathcal{X}_3$ are labeled, a donor is found from $\mathcal{X}_3^k$ for $\mathbf{x}_i \in \mathcal{X}_1^k$.

Once a donor is elected either from *Method 1* or from *Method 2*, the missing variable of $\mathbf{x}_i$ is imputed from the donor. *Method 2* solves smaller problems involving less number of data points for model fitting, but needs to train mixture models multiple times to impute all the missing variables in the dataset. On the contrary, *Method 1* solves a single large problem involving all data points.

Future research directions include finding ways of automatic domain information extraction. The construction of covariance matrices from incomplete data in the initialization of the EM algorithm is also an interesting problem. We expect that better covariance structure estimation, which will be available from better prior information, will be helpful for better replication of non-symmetric and non-elliptic cell subpopulations in the imputed results.

In the present study, we validated our method with lymphocyte data, where, for certain marker combinations, cell types tend to form relatively well-defined clusters. However, for other samples and marker combinations, clusters may be more elongated or less well-defined due to cells being at different stages of physiologic development. Fig. 16 indicates that flow cytometry clusters are often not Gaussian distributed. It may therefore be worth extending the ideas here to incorporate non-elliptical clusters using, for example, skewed Gaussian or skewed multivariate t components [7]. The cluster merging technique of [22] may also be helpful in this regard.

Appendix A. Derivation of EM algorithm for mixture of PPCA model with missing data

Suppose that we are given an incomplete observation set. We can divide each unit $\mathbf{x}_n$ as $\mathbf{x}_n = \begin{bmatrix} \mathbf{x}_n^{o_n} \\ \mathbf{x}_n^{m_n} \end{bmatrix}$ by separating the observed components and the missing components. Note that we do not assume that the observed variables come first and the missing variables next, and this should be understood as a notational convenience.

In the PPCA mixture model, the probability distribution of $\mathbf{x}$ is

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k p(\mathbf{x}|k)$$

where K is the number of components in the mixture and π_k is a mixing weight corresponding to the component density $p(\mathbf{x}|k)$. We estimate the set of unknown parameters $\theta = \{\pi_k, \boldsymbol{\mu}_k, \mathbf{W}_k, \sigma_k^2\}$ using an EM algorithm from the partial observations $\{\mathbf{x}_1^{o_1}, \cdots, \mathbf{x}_N^{o_N}\}$.

To develop an EM algorithm, we introduce indicator variables $\mathbf{z}_n = (z_{n1}, \cdots, z_{nK})$ for $n = 1, \cdots, N$. One and only one entry of $\mathbf{z}_n$ is nonzero, and $z_{nk} = 1$ indicates that the kth component is responsible for generating $\mathbf{x}_n$. We also include a set of the latent variables $\mathbf{t}_{nk}$ for each component and missing variables $\mathbf{x}_n^{m_n}$ to form the complete data $(\mathbf{x}_n^{o_n}, \mathbf{x}_n^{m_n}, \mathbf{t}_{nk}, \mathbf{z}_n)$ for $n = 1, \cdots, N$ and $k = 1, \cdots, K$. Then the corresponding complete data likelihood function has the form

$$\begin{aligned}
\mathcal{L}_C =& \sum_n \sum_k z_{nk} \ln \left[\pi_k p(\mathbf{x}_n, \mathbf{t}_{nk})\right] \\
=& \sum_n \sum_k z_{nk} \Bigg[\ln \pi_k - \frac{d}{2} \ln \sigma_k^2 - \frac{1}{2\sigma_k^2} \mathrm{tr}\left((\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T\right) \\
&+ \frac{1}{\sigma_k^2} \mathrm{tr}\left((\mathbf{x}_n - \boldsymbol{\mu}_k)\mathbf{t}_{nk}^T \mathbf{W}_k^T\right) - \frac{1}{2\sigma_k^2} \mathrm{tr}\left(\mathbf{W}_k^T \mathbf{W}_k \mathbf{t}_{nk} \mathbf{t}_{nk}^T\right) \Bigg],
\end{aligned}$$

where terms independent of the parameters are not included in the second equality. Instead of developing an EM algorithm directly on this likelihood function $\mathcal{L}_C$, we extend the strategy in [17] and build a two-stage EM algorithm, where each stage is a two-step process. This approach monotonically increases the value of the log-likelihood each round [17].

In the first stage of the two-stage EM algorithm, we update the component weight π_k and the component mean $\boldsymbol{\mu}_k$. We form a complete data log-likelihood function with the component indicator variables $\mathbf{z}_n$ and missing variables $\mathbf{x}_n^m$, while ignoring the latent variables $\mathbf{t}_{nk}$. Then we have the following likelihood function:

$$\begin{aligned}
\mathcal{L}_1 =& \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \ln[\pi_k p(\mathbf{x}_n^{o_n}, \mathbf{x}_n^{m_n}|k)] \\
=& \sum_n \sum_k z_{nk} \left[\ln \pi_k - \frac{1}{2} \ln |\mathbf{C}_k| - \frac{1}{2} \mathrm{tr}\left(\mathbf{C}_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T\right) \right]
\end{aligned}$$

where terms unrelated to the model parameters are omitted in the second line. We take the conditional expectation with respect to $p(\mathbf{z}_n, \mathbf{x}_n^{m_n}|\mathbf{x}_n^{o_n})$. Since the conditional probability factorizes as

$$p(\mathbf{z}_n, \mathbf{x}_n^{m_n}|\mathbf{x}_n^{o_n}) = p(\mathbf{z}_n|\mathbf{x}_n^{o_n})p(\mathbf{x}_n^{m_n}|\mathbf{z}_n, \mathbf{x}_n^{o_n}),$$

we have the following conditional expectations

$$\begin{aligned}
\langle z_{nk} \rangle =& p(k|\mathbf{x}_n^{o_n}) = \frac{\pi_k p(\mathbf{x}_n^{o_n}|k)}{\sum_{k'} \pi_{k'} p(\mathbf{x}_n^{o_n}|k')}, \\
\langle z_{nk}\mathbf{x}_n^{m_n} \rangle =& \langle z_{nk} \rangle \langle \mathbf{x}_n^{m_n} \rangle, \\
\langle \mathbf{x}_n^{m_n} \rangle =& \boldsymbol{\mu}_k^{m_n} + \mathbf{C}_k^{m_n o_n} {\mathbf{C}_k^{o_n o_n}}^{-1} (\mathbf{x}_n^{o_n} - \boldsymbol{\mu}_k^{o_n}), \\
\left\langle z_{nk}\mathbf{x}_n^{m_n}{\mathbf{x}_n^{m_n}}^T \right\rangle =& \langle z_{nk} \rangle \left\langle \mathbf{x}_n^{m_n}{\mathbf{x}_n^{m_n}}^T \right\rangle, \\
\left\langle \mathbf{x}_n^{m_n}{\mathbf{x}_n^{m_n}}^T \right\rangle =& \mathbf{C}_k^{m_n m_n} - \mathbf{C}_k^{m_n o_n} {\mathbf{C}_k^{o_n o_n}}^{-1} \mathbf{C}_k^{o_n m_n} + \langle \mathbf{x}_n^{m_n} \rangle \left\langle {\mathbf{x}_n^{m_n}}^T \right\rangle
\end{aligned}$$

where $\langle \cdot \rangle$ denotes the conditional expectation. Maximizing $\langle \mathcal{L}_1 \rangle$ with respect to π_k, using a Lagrange multiplier, and with respect to μ_k give the parameter updates

$$\widehat{\pi}_k = \frac{1}{N} \sum_n \langle z_{nk} \rangle, \tag{A.1}$$

$$\widehat{\boldsymbol{\mu}}_k = \frac{\sum_n \langle z_{nk} \rangle \begin{bmatrix} \mathbf{x}_n^{o_n} \\ \langle \mathbf{x}_n^{m_n} \rangle \end{bmatrix}}{\sum_n \langle z_{nk} \rangle}. \tag{A.2}$$

In the second stage, we include the latent variable $\mathbf{t}_{nk}$ as well to formulate the complete data log-likelihood function. The new values of $\widehat{\pi}_k$ and $\widehat{\boldsymbol{\mu}}_k$ are used in this step to compute sufficient statistics. Taking the conditional expectation on $\mathcal{L}_C$ with respect to $p(\mathbf{z}_n, \mathbf{t}_{nk}, \mathbf{x}_n^{m_n}|\mathbf{x}_n^{o_n})$, we have

$$\begin{aligned}
\langle \mathcal{L}_C \rangle = \sum_n \sum_k \langle z_{nk} \rangle \Bigg[& \ln \widehat{\pi}_k - \frac{d}{2} \ln \sigma_k^2 - \frac{1}{2\sigma_k^2} \mathrm{tr} \left(\left\langle (\mathbf{x}_n - \widehat{\boldsymbol{\mu}}_k)(\mathbf{x}_n - \widehat{\boldsymbol{\mu}}_k)^T \right\rangle \right) \\
& + \frac{1}{\sigma_k^2} \mathrm{tr} \left(\left\langle (\mathbf{x}_n - \widehat{\boldsymbol{\mu}}_k)\mathbf{t}_{nk}^T \right\rangle \mathbf{W}_k^T \right) - \frac{1}{2\sigma_k^2} \mathrm{tr} \left(\mathbf{W}_k^T \mathbf{W}_k \left\langle \mathbf{t}_{nk}\mathbf{t}_{nk}^T \right\rangle \right) \Bigg].
\end{aligned}$$

Since the the conditional probability factorizes

$$p(\mathbf{z}_n, \mathbf{t}_{nk}, \mathbf{x}_n^{m_n}|\mathbf{x}_n^{o_n}) = p(\mathbf{z}_n|\mathbf{x}_n^{o_n})p(\mathbf{x}_n^{m_n}|\mathbf{z}_n, \mathbf{x}_n^{o_n})p(\mathbf{t}_{nk}|\mathbf{z}_n, \mathbf{x}_n^{o_n}, \mathbf{x}_n^{m_n}),$$

we can evaluate the conditional expectations as follows :

$$\begin{aligned}
\left\langle (\mathbf{x}_n - \widehat{\boldsymbol{\mu}}_k)(\mathbf{x}_n - \widehat{\boldsymbol{\mu}}_k)^T \right\rangle =& \left(\begin{bmatrix} \mathbf{x}_n^{o_n} \\ \langle \mathbf{x}_n^{m_n} \rangle \end{bmatrix} - \widehat{\boldsymbol{\mu}}_k \right) \left(\begin{bmatrix} \mathbf{x}_n^{o_n} \\ \langle \mathbf{x}_n^{m_n} \rangle \end{bmatrix} - \widehat{\boldsymbol{\mu}}_k \right)^T \\
&+ \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{nk} \end{bmatrix}, \\
Q_{nk} =& \mathbf{C}_k^{m_n m_n} - \mathbf{C}_k^{m_n o_n} {\mathbf{C}_k^{o_n o_n}}^{-1} \mathbf{C}_k^{o_n m_n}, \\
\langle \mathbf{t}_{nk} \rangle =& \mathbf{M}_k^{-1} \mathbf{W}_k^T (\mathbf{x}_n - \widehat{\boldsymbol{\mu}}_k), \\
\left\langle (\mathbf{x}_n - \widehat{\boldsymbol{\mu}}_k)\mathbf{t}_{nk}^T \right\rangle =& \left\langle (\mathbf{x}_n - \widehat{\boldsymbol{\mu}}_k)(\mathbf{x}_n - \widehat{\boldsymbol{\mu}}_k)^T \right\rangle \mathbf{W}_k \mathbf{M}_k^{-1}, \\
\left\langle \mathbf{t}_{nk}\mathbf{t}_{nk}^T \right\rangle =& \mathbf{M}_k^{-1} \mathbf{W}_k^T \left\langle (\mathbf{x}_n - \widehat{\boldsymbol{\mu}}_k)(\mathbf{x}_n - \widehat{\boldsymbol{\mu}}_k)^T \right\rangle \mathbf{W}_k \mathbf{M}_k^{-1} \\
&+ \sigma_k^2 \mathbf{M}_k^{-1}.
\end{aligned}$$

Recall that the $q \times q$ matrix $\mathbf{M}_k = \mathbf{W}_k^T \mathbf{W}_k + \sigma_k^2 \mathbf{I}$. Then the maximization of $\langle \mathcal{L}_C \rangle$ with respect to $\mathbf{W}_k$ and σ_k^2 leads to the parameter updates,

$$\widehat{\mathbf{W}}_k = \left[\sum_n \langle z_{nk} \rangle \left\langle (\mathbf{x}_n - \widehat{\boldsymbol{\mu}}_k)\mathbf{t}_{nk}^T \right\rangle \right] \left[\sum_n \langle z_{nk} \rangle \left\langle \mathbf{t}_{nk}\mathbf{t}_{nk}^T \right\rangle \right]^{-1}, \tag{A.3}$$

$$\begin{aligned}
\widehat{\sigma}_k^2 =& \frac{1}{d \sum_n \langle z_{nk} \rangle} \Bigg[\sum_n \langle z_{nk} \rangle \operatorname{tr} \left(\left\langle (\mathbf{x}_n - \widehat{\boldsymbol{\mu}}_k)(\mathbf{x}_n - \widehat{\boldsymbol{\mu}}_k)^T \right\rangle \right) \\
&- 2 \sum_n \langle z_{nk} \rangle \operatorname{tr} \left(\left\langle (\mathbf{x}_n - \widehat{\boldsymbol{\mu}}_k)\mathbf{t}_{nk}^T \right\rangle \mathbf{W}_k^T \right) \\
&+ \sum_n \langle z_{nk} \rangle \operatorname{tr} \left(\mathbf{W}_k^T \mathbf{W}_k \left\langle \mathbf{t}_{nk}\mathbf{t}_{nk}^T \right\rangle \right) \Bigg].
\end{aligned} \tag{A.4}$$

Substituting the conditional expectations simplifies the M-step equations

$$\widehat{\mathbf{W}}_k = \mathbf{S}_k \mathbf{W}_k (\sigma_k^2 \mathbf{I} + \mathbf{M}_k^{-1} \mathbf{W}_k^T \mathbf{S}_k \mathbf{W}_k)^{-1}, \tag{A.5}$$

$$\widehat{\sigma}_k^2 = \frac{1}{d} \operatorname{tr} \left(\mathbf{S}_k - \mathbf{S}_k \mathbf{W}_k \mathbf{M}_k^{-1} \widehat{\mathbf{W}}_k^T \right) \tag{A.6}$$

where

$$\mathbf{S}_k = \frac{1}{N \widehat{\pi}_k} \sum_n \langle z_{nk} \rangle \left\langle \left(\begin{bmatrix} \mathbf{x}_n^{o_n} \\ \langle \mathbf{x}_n^{m_n} \rangle \end{bmatrix} - \widehat{\boldsymbol{\mu}}_k \right) \left(\begin{bmatrix} \mathbf{x}_n^{o_n} \\ \langle \mathbf{x}_n^{m_n} \rangle \end{bmatrix} - \widehat{\boldsymbol{\mu}}_k \right)^T \right\rangle.$$

Each iteration of the EM algorithm updates the set of old parameters $\{\pi_k, \boldsymbol{\mu}_k, \mathbf{W}_k, \sigma_k^2\}$ with the set of new parameters $\{\widehat{\pi}_k, \widehat{\boldsymbol{\mu}}_k, \widehat{\mathbf{W}}_k, \widehat{\sigma}_k^2\}$ in (A.1), (A.2), (A.5) and (A.6). The algorithm terminates when the value of the log-likelihood function changes less than a predefined accuracy constant.

ID	N_1	N_2	N_e
Patient1	10000	10000	5223
Patient2	7000	7000	4408
Patient3	3000	3000	3190

Table B.7: Datasets from three patients in the single tube experiments (Section 4.1). Each tube is divided into two data files and an evaluation set. N_1 and N_2 denote the sizes of the two data files, and N_e is the size of the evaluation set.

Appendix B. Computing KL divergences

For each experiment in Section 4, we quantify the imputation results using the KL divergence.

Appendix B.1. Single tube experiments

In the single tube experiments, each dataset corresponding to the different patients is divided into two data files and a separate evaluation set. Table B.7 summarizes the cell counts in these sets. N_1, N_2 and N_e are the cell counts of the two files and the hold-out set, respectively. After imputing the two files with either the NN or the Cluster-NN method, the KL divergences are computed. The empirical estimate of the KL divergence is

$$\begin{aligned} KL(g \,\|\, f) =& \mathbb{E}_g \left[\log g - \log f\right] \\ \approx& \frac{1}{N_e} \sum_{n=1}^{N_e} \left[\log g\left(\widehat{\mathbf{x}}_n\right) - \log f\left(\widehat{\mathbf{x}}_n\right)\right] \\ \approx& \frac{1}{N_e} \sum_{n=1}^{N_e} \left[\log \widehat{g}\left(\widehat{\mathbf{x}}_n\right) - \log \widehat{f}\left(\widehat{\mathbf{x}}_n\right)\right] \end{aligned}$$

where the distributions f and g are replaced by their corresponding density estimates and the expectation is approximated by a finite sum over imputed results $\widehat{\mathbf{x}}_n$ on the hold-out set of size N_e. For $\widehat{f}$ and $\widehat{g}$, we used kernel density estimation on the ground truth data and the imputed data, respectively.

Appendix B.2. Multiple tube experiments

As explained in Section 4.2, three tubes per patient are available in the multiple tube experiments. The third tube of higher dimension is a reference dataset and is not involved during the file matching. Each of the two lower dimensional tubes is split into two halves. The first halves of the two tubes

ID	N_1	N_2	N_{e1}	N_{e2}	N_3
Patient1	10000	10000	3982	21828	47248
Patient2	8000	8000	14661	3793	28101
Patient3	2000	5000	1817	7228	9795

Table B.8: Datasets from three patients in the multiple tube experiments (Section 4.2). N_1 and N_2 denote the sizes of the two data files, and N_{e1} and N_{e2} denote the sizes of the evaluation sets. N_3 is the number of cells in the additional tube that is treated as the ground truth.

form the incomplete data: file 1 and file 2 with N_1 and N_2 cells, respectively. The second halves of size N_{e1} and N_{e2} form the evaluation sets and their imputed results are used to approximate the expectation of the KL divergence. For each patient, the sizes of these sets are shown in Table B.8. The reason for splitting each tube in half is so that the data used to approximate the expectation are independent to the data used to estimate density of the imputed result. Therefore, the imputation result of file 1 is evaluated by

$$KL(g_1 \,\|\, f) \approx \frac{1}{N_{e1}} \sum_{n=1}^{N_{e1}} \left[\log \widehat{g}_1 \left(\widehat{\mathbf{x}}_n \right) - \log \widehat{f} \left(\widehat{\mathbf{x}}_n \right) \right]$$

where $\widehat{g}_1$ is the kernel density estimate based on imputed rows from the first half of tube 1. The third tube is treated as the ground truth data and used to obtain the density estimate $\widehat{f}$.

When evaluating the KL divergence of file 2, $\widehat{g}_1$ is replaced by $\widehat{g}_1$, the kernel density estimate on the imputed result of file 2, and the finite sum is taken over the evaluation set of size N_{e2}.

References

[1] H. Shapiro, Practical Flow Cytometry, 3rd Edition, Wiley-Liss, 1994.

[2] M. Brown, C. Wittwer, Flow cytometry: Principles and clinical applications in hematology, Clinical Chemistry 46 (2000) 1221–1229.

[3] C. E. Pedreira, E. S. Costa, S. Barrena, Q. Lecrevisse, J. Almeida, J. J. M. van Dongen, A. Orfao, Generation of flow cytometry data files with a potentially infinite number of dimensions, Cytometry A 73A (2008) 834–846.

[4] M. J. Boedigheimer, J. Ferbas, Mixture modeling approach to flow cytometry data, Cytometry Part A 73 (2008) 421 – 429.

[5] C. Chan, F. Feng, J. Ottinger, D. Foster, M. West, T. Kepler, Statistical mixture modeling for cell subtype identification in flow cytometry, Cytometry Part A 73 (2008) 693–701.

[6] K. Lo, R. R. Brinkman, R. Gottardo, Automated gating of flow cytometry data via robust model-based clustering, Cytometry Part A 73 (2008) 321 – 332.

[7] S. Pyne, X. Hu, K. Wang, E. Rossin, T.-I. Lin, L. M. Maier, C. Baecher-Allan, G. J. McLachlan, P. Tamayo, D. A. Hafler, P. L. D. Jager, J. P. Mesirov, Automated high-dimensional flow cytometric data analysis, PNAS 106 (2009) 8519–8524.

[8] J. Lakoumentas, J. Drakos, M. Karakantza, G. C. Nikiforidis, G. C. Sakellaropoulos, Bayesian clustering of flow cytometry data for the diagnosis of B-chronic lymphocytic leukemia, Journal of Biomedical Informatics 42 (2009) 251–261.

[9] K. Carter, R. Raich, W. Finn, A. O. Hero, Information preserving component analysis: data projections for flow cytometry analysis, Journ. of Selected Topics in Signal Processing 3 (2009) 148–158.

[10] K. Carter, R. Raich, W. Finn, A. O. Hero, Fine: Fisher information nonparametric embedding, IEEE Trans. on Pattern Analysis and Machine Intelligence 31 (2009) 2093–2098.

[11] S. Perfetto, P. K. Chattopadhyay, M. Roederer, Seventeen-colour flow cytometry: unravelling the immune system, Nature Reviews Immunology 4 (2004) 648–655.

[12] M. Sánchez, J. Almeida, B. Vidriales, M. López-Berges, M. García-Marcos, M. Moro, A. Corrales, M. Calmuntia, J. S. Miguel, A. Orfao, Incidence of phenotypic aberrations in a series of 467 patients with B chronic lymphoproliferative disorders: basis for the design of specific four-color stainings to be used for minimal residual disease investigation, Leukemia 16 (2002) 1460–1469.

[13] R. Little, D. Rubin, Statistical analysis with missing data, 2nd Edition, Wiley, 2002.

[14] S. Rässler, Statistical matching: a frequentist theory, practical applications, and alternative Bayesian approaches, no. 168 in Lecture Notes in Statistics, Springer, 2002.

[15] M. Tipping, C. Bishop, Probabilistic principal component analysis, Journal of the Royal Statistical Society, B 6 (1999) 611–622.

[16] C. Fraley, A. E. Raftery, How many clusters? Which clustering method? Answers via model-based cluster analysis, The Computer Journal 41 (1998) 578–588.

[17] M. Tipping, C. Bishop, Mixtures of probabilistic principal component analysis, Neural Computation 11 (1999) 443–482.

[18] Z. Ghahramani, M. Jordan, Supervised learning from incomplete data via an EM approach, Advances in Neural Information Processing Systems 6 (1994) 120–127.

[19] H. Zola, B. Swart, I. Nicholson, B. Aasted, A. Bensussan, L. Boumsell, C. Buckley, G. Clark, K. Drbal, P. Engel, D. Hart, V. Horejsi, C. Isacke, P. Macardle, F. Malavasi, D. Mason, D. Olive, A. Saalmueller, S. F. Schlossman, R. Schwartz-Albiez, P. Simmons, T. F. Tedder, M. Uguccioni, H. Warren, CD molecules 2005: Human cell differentiation molecules, Blood 106 (2005) 3123–3126.

[20] 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, R. Gentleman, R. R. Brinkman, Per-channel basis normalization methods for flow cytometry data, Cytometry Part A 77A (2010) 121–131.

[21] G. Finak, J.-M. Perez, A. Weng, R. Gottardo, Optimizing transformations for automated, high throughput analysis of flow cytometry data, BMC Bioinformatics 11 (2010) 546. doi:10.1186/1471-2105-11-546.

[22] G. Finak, A. Bashashati, R. Brinkman, R. Gottardo, Merging mixture components for cell population identification in flow cytometry, Advances in Bioinformatics 2009. doi:10.1155/2009/247646.