2023 IEEE 9th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP) | 979-8-3503-4452-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/CAMSAP58249.2023.10403522

Probability Mass Function Estimation Approaches with Application to Flow Cytometry Data Analysis

Philippe Flores[2,3], Joseph K. Chege[1], Konstantin Usevich[2,4], Martin Haardt[1], Arie Yeredor[5], and David Brie[2,3]
[1]*Communications Research Laboratory, Ilmenau University of Technology*
[2]*Centre de Recherche en Automatique de Nancy*, [3]*Université de Lorraine*, [4]*CNRS*
[5]*School of Electrical Engineering, Tel Aviv University*
{joseph.chege, martin.haardt}@tu-ilmenau.de, {philippe.flores, konstantin.usevich, david.brie}@univ-lorraine.fr,
ariey@tauex.tau.ac.il

***Abstract*—In this paper, we study three recently proposed probability mass function (PMF) estimation methods for flow cytometry data analysis. By modeling the PMFs as a mixture of simpler distributions, we can reformulate the PMF estimation problem as three different tensor-based approaches: a least squares coupled tensor factorization approach, a least squares partially coupled tensor factorization approach, and a Kullback-Leibler divergence (KLD)-based expectation-maximization (EM) approach. In the coupled methods, the full PMF is estimated from lower-order empirical marginal distributions, while the EM approach estimates the full PMF directly from the observed data. The three approaches are evaluated in the context of simulated and real data experiments.**

***Index Terms*—Probability Mass Functions (PMF), Naïve Bayes Model, Low-Rank Tensor Decomposition, Flow Cytometry**

I. Introduction

Flow cytometry (FCM) is one of the most popular techniques for biological cell analysis. It is the reference technique in immunology because it allows for the identification of rare cell populations and thus improves the knowledge of the human immune system [1]. From a data analysis point of view, a cytometer produces a point cloud in an N-dimensional space, where each point measured represents N characteristics called markers. The aim is to identify the different cell populations in this set of data points. Conventional analysis carried out manually by practitioners essentially consists of a series of 2-dimensional analyses; it becomes complex, subjective, and costly in terms of manpower and time when N increases. This has motivated the development of automatic methods [2], [3], which are still costly and difficult to apply to large data sets. Furthermore, these methods have a limited performance for the analysis of rare cell populations, and their associated visualization tools are often difficult to interpret by end-users.

Recently, probabilistic approaches based on the estimation of the joint density of the data have been explored. Estimating a probability mass function (PMF) is a challenging problem in practice due to the curse of dimensionality: the amount of data required to provide an accurate estimate increases exponentially with the dimension N of the problem. To cope with the curse of dimensionality, the methods presented adopt a naïve Bayes model of the joint density whose complexity remains linear with N. Under this model, estimating the N-dimensional PMF can be reduced to estimating the factors of a CP (canonical polyadic) tensor model [4]. To estimate the factors from a set of T observations, the authors in [5] proposed a cost function using the Frobenius norm that coupled lower-order marginals. While this approach works quite effectively, the complexity is a function of the number of lower-order marginals. For example, the number of order-3 marginals is $\binom{N}{3}$ and can be quite large for large N. Therefore, to reduce the complexity of [5], a partially coupled tensor factorization that only considers a subset of order-3 marginals was proposed in [6]. On the other hand, [7] and [8] proposed an expectation-maximization (EM) algorithm which estimates the full PMF tensor directly from the data, obviating the need to compute lower-order marginals.

While the coupled methods are based on the least squares criterion, the similarity criterion in the EM approach is the Kullback-Leibler divergence (KLD) between the estimated and the (unknown) true PMF tensor. These two criteria behave quite differently in terms of how they handle rare events, which have small empirical probabilities. The least squares criterion, being a symmetric distance measure, may assign an extremely small (or even zero) probability to a rare event because it is penalized equally for larger and smaller probabilities. On the other hand, the KLD criterion, being an asymmetric distance measure, would not allow zero probability where the empirical probability is nonzero, even if that probability is very small. In fact, if zero probability were assigned in this case, the KLD would diverge to infinity. As was demonstrated in [9], the KLD criterion results in more accurate PMF estimates than the least squares criterion. Therefore, in the context of FCM data analysis, the EM algorithm is expected to detect small cell populations more accurately than the least squares-based methods.

In this paper, we compare the performance of three recently proposed PMF estimation schemes: least squares fully coupled and partially coupled tensor factorizations [6], [10] and the EM algorithm [7], [8], which is based on the KLD criterion. In particular, we are interested in the conditions under which the three algorithms detect small populations as well as the computational efficiency of the algorithms. To compare these methods, we examine both synthetic data as well as real flow cytometry data.

II. Naïve Bayes Model for PMF Estimation

Let $\mathbf{x} = \left(X^{(1)}, \ldots, X^{(N)}\right)$ be a random vector taking values in $\mathcal{I}^{(1)} \times \cdots \times \mathcal{I}^{(N)}$ where $\mathcal{I}^{(n)} = [x_{\min}^{(n)}, x_{\max}^{(n)}]$. We assume that the T rows $\mathbf{x}_t$ of $\mathbf{X}$ are realizations of the random vector $\mathbf{x}$. Our goal is to estimate the multivariate probability density function (PDF) $\mathrm{p}(\mathbf{x}) = \mathrm{p}(X^{(1)}, \ldots, X^{(N)})$ from the observation matrix $\mathbf{X}$. One first approach is to consider an N-dimensional histogram. In this case, each interval $\mathcal{I}^{(n)}$ is separated in I equal bins from $\Delta_1^{(n)} = [x_0^{(n)}, x_1^{(n)}]$ to $\Delta_I^{(n)} = [x_{I-1}^{(n)}, x_I^{(n)}]$, where $x_0^{(n)} = x_{\min}^{(n)}$ and $x_I^{(n)} = x_{\max}^{(n)}$. The histogram $\boldsymbol{\mathcal{H}} \in (\mathbb{R}^I)^N$ is an order-N tensor which can be

This work is partially supported by the German Research Foundation (DFG) within the PROMETHEUS project (reference no. HA 2239/16-1, project no. 462458843).

interpreted as the discretized joint PDF:

$$\begin{aligned}\boldsymbol{\mathcal{H}}_{i_1\dots i_N} &= \Pr(\mathbf{x} \in \Delta_{i_1}^{(1)} \times \cdots \times \Delta_{i_N}^{(N)}) \\ &= \int_{\Delta_{i_1}^{(1)}} \cdots \int_{\Delta_{i_N}^{(N)}} \mathrm{p}(\mathbf{x}) dX^{(1)} \cdots dX^{(N)}\end{aligned} \tag{1}$$

A naïve approach to estimate the histogram from $\mathbf{X}$ is to count the number of samples $\mathbf{x}_t$ in each N-dimensional bin:

$$\widetilde{\boldsymbol{\mathcal{H}}}_{i_1\cdots i_N} = \frac{1}{T} \operatorname{Card}\left\{ t \in [\![1,T]\!] \,\middle|\, \mathbf{x}_t \in \Delta_{i_1}^{(1)} \times \cdots \times \Delta_{i_N}^{(N)} \right\}. \tag{2}$$

However, it requires a number of samples growing exponentially with N. To give some figures, with $N = 8$ and $I = 20$, $\boldsymbol{\mathcal{H}}$ is described by $I^N \approx 10^{10}$ values, and a prohibitively large number of samples is required to produce an accurate estimate. This drawback is referred to as the curse of dimensionality. To cope with it, [5] proposed to use a model whose complexity remains linear with N [10]. The naïve Bayes model (NBM) [5] introduces a latent variable L taking values in $\{1, \dots, R\}$, such that $X^{(n)}$ is conditionally independent on L:

$$\mathrm{p}(\mathbf{x}) = \sum_{r=1}^{R} \Pr(L = r) \prod_{n=1}^{N} \mathrm{p}(X^{(n)} | L = r). \tag{3}$$

By inserting (3) into (1), the NBM corresponds to an order-N canonical polyadic decomposition (CPD) [11] of $\boldsymbol{\mathcal{H}}$:

$$\begin{aligned}\boldsymbol{\mathcal{H}}_{i_1\cdots i_N} &= \sum_{r=1}^{R} \Pr(L = r) \prod_{n=1}^{N} \Pr\left(X^{(n)} \in \Delta_{i_n}^{(n)} \middle| L = r\right) \\ \boldsymbol{\mathcal{H}} &= \left[\!\!\left[\boldsymbol{\lambda}; \mathbf{A}^{(1)}, \dots, \mathbf{A}^{(N)} \right]\!\!\right] = \sum_{r=1}^{R} \boldsymbol{\lambda}_r \mathbf{a}_r^{(1)} \circ \cdots \circ \mathbf{a}_r^{(N)}\end{aligned} \tag{4}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^R$ contains the probabilities $\Pr(L = r)$, and the factors $\mathbf{a}_r^{(n)}$ are 1D conditional marginals (i.e., the values of $\Pr(X^{(n)} \in \Delta_{i_n}^{(n)} | L = r)$). Thus, R corresponds to the tensor rank of $\boldsymbol{\mathcal{H}}$. As the factor matrices $\mathbf{A}^{(n)} = \begin{bmatrix} \mathbf{a}_r^{(1)} & \dots & \mathbf{a}_r^{(R)} \end{bmatrix}$ and $\boldsymbol{\lambda}$ represent probabilities, they should satisfy non-negativity constraints ($\boldsymbol{\lambda} \geq 0$, $\mathbf{A}^{(n)} \geq 0$), and sum-to-one constraints ($\mathbb{1}^\mathsf{T}\boldsymbol{\lambda} = 1, \mathbb{1}^\mathsf{T}\mathbf{A}^{(n)} = \mathbb{1}^\mathsf{T}$).

III. Overview of algorithms

A. Least squares coupling of 3D marginals

The coupled tensor factorization of 3D marginals is based on the fact that a marginalized NBM (3) is a lower-order NBM. Indeed, due to simplex constraints, the 3D histogram $\boldsymbol{\mathcal{H}}^{(jk\ell)}$ of a subset of variables $(X^{(j)}, X^{(k)}, X^{(\ell)})$ has the CPD

$$\boldsymbol{\mathcal{H}}^{(jk\ell)} = \left[\!\!\left[\boldsymbol{\lambda}; \mathbf{A}^{(j)}, \mathbf{A}^{(k)}, \mathbf{A}^{(\ell)} \right]\!\!\right]. \tag{5}$$

The 3D histograms for all triplets of variables $\{j, k, \ell\}$ are estimated with

$$\widetilde{\mathcal{H}}_{i_j i_k i_\ell}^{(jk\ell)} = \frac{1}{T} \operatorname{Card}\left\{ \mathbf{x}_t \in \Delta_{i_j}^{(j)} \times \Delta_{i_k}^{(k)} \times \Delta_{i_\ell}^{(\ell)} \right\} \tag{6}$$

which are easily computable compared to the full N-D histogram (2). Estimating the factors comes to solving the following coupled tensor optimization problem:

$$\begin{aligned}&\widehat{\boldsymbol{\lambda}}, \widehat{\boldsymbol{A}}^{(1)}, \dots, \widehat{\boldsymbol{A}}^{(N)} = \underset{\boldsymbol{\lambda}, \mathbf{A}^{(1)}, \dots, \mathbf{A}^{(N)}}{\operatorname{argmin}} \\ &\sum_{j=1}^{N-2} \sum_{k=j+1}^{N-1} \sum_{\ell=k+1}^{N} \left\| \widetilde{\boldsymbol{\mathcal{H}}}^{(jk\ell)} - \left[\!\!\left[\boldsymbol{\lambda}; \mathbf{A}^{(j)}, \mathbf{A}^{(k)}, \mathbf{A}^{(\ell)} \right]\!\!\right] \right\|_F^2 \\ &\text{s.t. } \boldsymbol{\lambda} \geq 0, \mathbf{A}^{(n)} \geq 0, \mathbb{1}^\mathsf{T}\boldsymbol{\lambda} = 1, \mathbb{1}^\mathsf{T}\mathbf{A}^{(n)} = \mathbb{1}^\mathsf{T}.\end{aligned} \tag{7}$$

This method is referred to as Coupled Tensor Factorization or CTF3D, which was initially proposed in [5] and is solved via an alternating optimization procedure using the alternating direction method of multipliers (ADMM) [12].

B. Least squares partial coupling of 3D marginals

As the number of dimensions N increases, the number of triplets in (7) is $\binom{N}{3}$ and therefore increases cubically with N. In practice, this complexity can lead to computational issues. For example, with $N = 20$ variables, 1140 3D marginals must be estimated, stored and handled to solve (7). To reduce the complexity of the approach [5], we showed in [6] that it is not necessary to consider all possible triplets like in (7). This leads to the following optimization problem:

$$\begin{aligned}&\underset{\boldsymbol{\lambda}, \mathbf{A}^{(1)}, \dots, \mathbf{A}^{(N)}}{\operatorname{argmin}} \sum_{\{j,k,\ell\} \in \mathcal{T}} \left\| \widetilde{\boldsymbol{\mathcal{H}}}^{(jk\ell)} - \left[\!\!\left[\boldsymbol{\lambda}; \mathbf{A}^{(j)}, \mathbf{A}^{(k)}, \mathbf{A}^{(\ell)} \right]\!\!\right] \right\|_F^2 \\ &\text{s.t. } \boldsymbol{\lambda} \geq 0, \mathbf{A}^{(n)} \geq 0, \mathbb{1}^\mathsf{T}\boldsymbol{\lambda} = 1, \mathbb{1}^\mathsf{T}\mathbf{A}^{(n)} = \mathbb{1}^\mathsf{T},\end{aligned} \tag{8}$$

where $\mathcal{T}$ is the subset of triplets considered in the coupling. For example, if $\mathcal{T} = \{\{j, k, \ell\} \subset [\![1, N]\!] \,|\, j < k < \ell\}$, then (8) is equivalent with CTF3D. This approach will be denoted as Partial Coupled Tensor Factorization of 3D marginals or PCTF3D. Some possible coupling strategies are presented in [6]. In the following, the coupling $\mathcal{T}$ will be chosen randomly such that $\mathcal{T}$ contains a random half of all possible triplets.

C. Expectation-maximization (EM) algorithm

Instead of estimating $\boldsymbol{\mathcal{H}}$ from lower-order marginals, we can directly use the observations $\mathbf{X}$ to compute the maximum-likelihood (ML) estimate of $\boldsymbol{\mathcal{H}}$. This is achieved via the expectation-maximization (EM) algorithm [7]. It can be shown that maximizing the log-likelihood of $\mathbf{X}$ given the CPD model of $\boldsymbol{\mathcal{H}}$ is equivalent to minimizing the Kullback-Leibler divergence (KLD) between the true PMF and the model. As was demonstrated in [9], the KLD criterion is a more appropriate choice for the PMF estimation task as compared to the least squares criterion. The optimization problem for maximizing the log-likelihood of $\mathbf{X}$ is given by

$$\begin{aligned}&\underset{\boldsymbol{\lambda}, \mathbf{A}^{(1)}, \dots, \mathbf{A}^{(N)}}{\operatorname{argmin}} -\sum_{t=1}^{T} \log \sum_{r=1}^{R} \boldsymbol{\lambda}_r \prod_{n=1}^{N} \mathbf{a}_{\mathbf{x}_{n,t},r}^{(n)} \\ &\text{s.t. } \boldsymbol{\lambda} \geq 0, \mathbf{A}^{(n)} \geq 0, \mathbb{1}^\mathsf{T}\boldsymbol{\lambda} = 1, \mathbb{1}^\mathsf{T}\mathbf{A}^{(n)} = \mathbb{1}^\mathsf{T},\end{aligned} \tag{9}$$

where $\mathbf{x}_{n,t}$ is the n-th element of the observed vector $\mathbf{x}_t$ and $\mathbf{a}_{i,r}^{(n)}$ represents the i-th element of $\mathbf{a}_r^{(n)}$.

It turns out that the NBM conveniently lends itself to the EM algorithm. Since, according to the model, each observed data vector $\mathbf{x}_t$ depends on the corresponding (unobserved) latent state $s_t \in \{1, \dots, R\}$ (s_t is a realization of the latent variable L), we can define the complete data vector $\mathbf{z}_t = [\mathbf{x}_t^\mathsf{T}, s_t]^\mathsf{T}$. The EM algorithm then consists of two steps. In the E-Step, the *a posteriori* distribution of the latent variable L given the current observations $\mathbf{X}$ and parameters $\boldsymbol{\theta} = \left\{ \boldsymbol{\lambda}, \mathbf{A}^{(1)}, \dots, \mathbf{A}^{(N)} \right\}$ is computed. Since the latent states are unobserved, the distribution is approximated as the expected value of the complete data given the observations. With $\boldsymbol{\theta}'$ as the initial setting of the parameters, we compute

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}') = \mathbb{E}[\log \Pr(\{\mathbf{z}_t\}_{t=1}^T; \boldsymbol{\theta}) \,|\, \{\mathbf{x}_t\}_{t=1}^T; \boldsymbol{\theta}']. \tag{10}$$

In the M-Step, the parameter values that maximize $Q(\boldsymbol{\theta}, \boldsymbol{\theta}')$ are computed, i.e.,

$$\boldsymbol{\theta}' \leftarrow \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}'). \tag{11}$$

These two steps are then iterated until convergence. The maximizing solutions admit a closed form consisting of simple divisions [7], i.e.,

$$\boldsymbol{\lambda}_r = \frac{C_r}{\sum_{g=1}^{R} C_g} \quad \text{and} \quad \mathbf{a}_{i,r}^{(n)} = \frac{K_{i,r}^{(n)}}{\sum_{j=1}^{I_n} K_{j,r}^{(n)}}, \tag{12}$$

where $C_r = \sum_{t=1}^{T} c_{t,r}(\boldsymbol{\theta}')$, $K_{i,r}^{(n)} = \sum_{t:\mathbf{x}_{n,t}=i} c_{t,r}(\boldsymbol{\theta}')$, and $c_{t,r}(\boldsymbol{\theta}') = \Pr(s_t = r \,|\, \mathbf{x}_t; \boldsymbol{\theta}')$.

While the EM algorithm is simple to implement and has fast, closed-form updates, it is known to exhibit slow convergence. Thus, in the following, we employ the SQUAREM-PMF algorithm, proposed in [8] to accelerate the convergence of the EM algorithm.

D. Complexity analysis

The EM algorithm consists of three procedures: the E-Step, the M-Step, and the log-likelihood computation to check for convergence. The E-Step and the M-Step are based on the coefficients $c_{t,r}(\boldsymbol{\theta}')$, C_r, and $K_{i,r}^{(n)}$ (cf. Section III-C), whose complexities are $\mathcal{O}(TNR)$, $\mathcal{O}(TR)$, and $\mathcal{O}(TR)$, respectively. Furthermore, the complexity of the log-likelihood computation (cf. (9)) is $\mathcal{O}(T(N+R))$. Therefore, the overall computation complexity is given by $\mathcal{O}(TNR)$. The SQUAREM-PMF algorithm [8] consists of three EM updates, some acceleration procedures, and the log-likelihood computation. However, the complexity of SQUAREM-PMF is dominated by the EM updates and the log-likelihood computation; hence, it is slightly higher than that of the EM algorithm.

Concerning least squares methods, the complexity of both PCTF3D and CTF3D relies on the number of marginals to compute and couple. The number of triplets for CTF3D is $\binom{N}{3}$ which is in $\mathcal{O}(N^3)$. In this paper, the coupling strategy used for PCTF3D consists in taking randomly half of all possible triplets. Therefore, the number of triplets for PCTF3D is also in $\mathcal{O}(N^3)$. Therefore, PCTF3D and CTF3D have the same asymptotic complexity even if PCTF3D's complexity is half of CTF3D's. The complexity of the least squares methods is then computed by adding the complexity of the computation of 3D histograms with the complexity of the internal ADMM [13] which gives a complexity in $\mathcal{O}(N^3T + N^3RI^2(R+I))$. In practice, the computations are dominated by ADMM which yields a complexity in $\mathcal{O}(N^3RI^2(R+I))$. However, it is possible that the computations of the 3D marginals become dominant, especially for large data sets.

IV. Numerical Experiments

A. Small cluster sensitivity experiment

In flow cytometry, end users search for small cell populations. Thus, we propose an experiment to test the sensitivity of the methods presented in Section III. To do this, $R_{\text{th}} = 3$ multivariate discrete Gaussian random variables were generated with $N = 7$ and $I = 20$. For each theoretical distribution, $T = 10^5$ samples were generated with 8 different proportions: $\lambda_1 \in \{0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, \frac{1}{3}\}$, $\lambda_2 = \frac{1}{3}$ and $\lambda_3 = 1 - (\lambda_1 + \lambda_2)$. With theoretical datasets generated, the three methods were applied with increasing rank until the smallest component was part of the estimate. Therefore, this procedure stopped if there existed an estimated CPD rank-one term such that $\prod_{n=1}^{N} \|\widehat{\mathbf{a}}_r^{(n)} - \mathbf{a}_r^{(n)}\|_2 < 10^{-4}$. Figure 1 shows the minimum rank required for each experiment averaged over 100 trials. It can be seen that, compared to CTF3D and PCTF3D, SQUAREM-PMF is more sensitive to small clusters as R must be higher if one wants to obtain a small component with (P)CTF3D. Unlike CTF3D and PCTF3D, for SQUAREM-PMF, the estimation rank can be chosen close to the expected number of clusters. In terms of computational load, SQUAREM-PMF is faster on harder problems because the smallest rank-one term is found with $R = 3$. As λ_1 increases, the minimum rank required is close to R_{th} for all methods and thus SQUAREM-PMF is slower than the other methods.

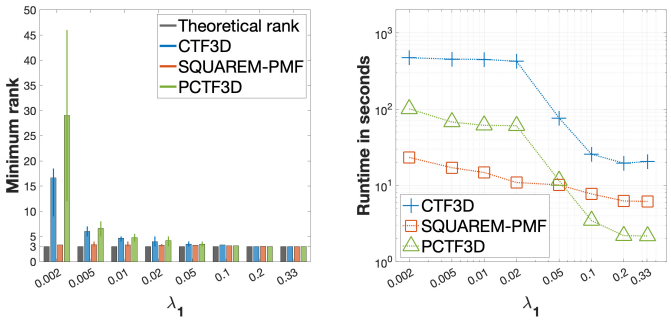

Fig. 1. Results for the sensitivity experiment. 100 trials were run on datasets with $N = 7$ variables featuring $R_{th} = 3$ theoretical rank-one terms. **Left plot:** Mean value over 100 trials of the minimal rank that provides the desired rank-one term. **Right plot:** Median value over 100 trials of the runtime in seconds.

B. Runtime analysis

The three algorithms which we examine have varying complexities. For example, the computational time of CTF3D and PCTF3D is less dependent on T in comparison with SQUAREM-PMF. In order to verify this empirically, we ran three experiments on each method and averaged the results over 100 trials. The performance criteria were the runtime and the factor match score (FMS) [14].

a) Evolution with respect to N: In the first experiment, $T = 10^5$ samples were generated from random factors with $N \in [\![4, 10]\!]$, $I \in \{5, 10, 15\}$ and $R = 5$. For each setting, the three proposed methods were run to obtain a CPD of rank $R = 5$. The left plots of Figure 2 show that CTF3D's runtimes increase with N, and CTF3D becomes slower than SQUAREM-PMF for the highest values of N (note that the total number of elements in 3D histograms becomes comparable with the number of data points T). The runtimes for SQUAREM-PMF do not depend strongly on I.

b) Evolution with respect to R: Next, for 100 random observation matrices of size $(T = 10^4) \times (N = 5)$, the proposed methods were run to obtain a CPD with different ranks $R \in [\![3, 20]\!]$. The middle plots of Figure 2 shows that the runtimes increase with R. For small values of R, SQUAREM-PMF converges with less iterations and thus has a lower computation time for lower ranks; for higher ranks CTF3D and PCTF3D are faster. For CTF3D and PCTF3D, we observed a drop of the runtime for $I = 4$ and $R > 20$ (not shown), which is explained by a loss of identifiability after those ranks.

c) Evolution with respect to T: Finally, to study the complexity of T, the proposed methods were applied on 100 datasets with $N = 6$ to obtain a rank $R = 5$ CPD. The right plots of Figure 2 show that the runtime of CTF3D does not depend on T, while achieving similar performance to that of SQUAREM-PMF. However, SQUAREM-PMF's runtimes increase considerably with T (some trials took a few hours to run).

V. Application to Flow Cytometry

A. Flow cytometry data analysis

One main flow cytometry (FCM) data analysis problem is the search of small cell populations. In practice, manual gating permits to

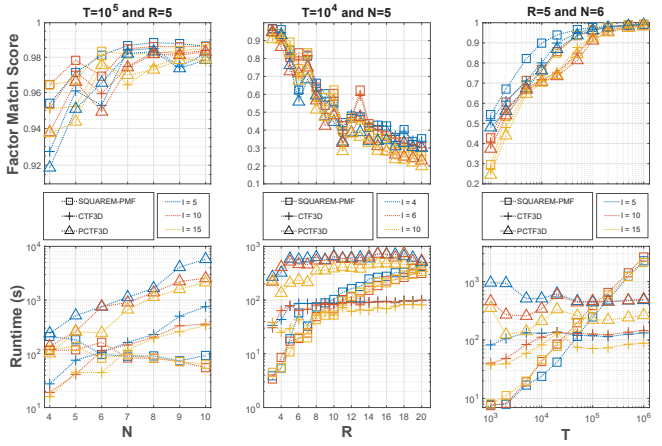

Fig. 2. Runtime and FMS [14] depending on 4 main complexity parameters N, T, R and I. **Left plots:** N experiment with $N \in [\![4, 10]\!]$. **Middle plots:** R experiment with $R \in [\![3, 20]\!]$. **Right plots:** experiment regarding T.

find as little cell populations as possible, at a cost of subjective and time-consuming analysis. Moreover, as soon as $N \geq 3$, assessing an overview of all cells while considering all possible markers is challenging. For example, viSNE [3] provides a dataset overview by projecting data onto a 2D map; whereas SPADE [2] uses a minimum spanning tree to present a visualization of k-means clusters.

B. Connection to the data model

With N markers for T cells, FCM data can be interpreted as a $T \times N$ matrix $\mathbf{X}$, so that each row $\mathbf{x}_t$ contains the characteristics of the t-th cell. Typical values for T are from thousands to millions of cells while N ranges from 1 to 30. By considering the N markers as a random vector, the PDF of this random vector is modelled with the NBM as presented in Section II. This model can be interpreted as a sum where the r-th component represents a cell population. The factor $\mathbf{a}_r^{(n)}$ then represents the expression of the n-th marker for the r-th component while λ_r is the proportion of cells such that $\lambda_r N$ cells are contained in the r-th component.

C. Real data experiment: controlled dataset

To validate our methods on real data, flow cytometry datasets were created in a controlled environment. Three cell lines were considered: Lymphocytes B and T (LB), Lymphocytes T (LT) and Macrophages (MP), having different responses according to $N = 4$ markers (see Table I). Cells were then mixed in different proportions resulting in 3 datasets with $T = 10^5$ cells, where the MP proportion varies (around 20%, 8% and 1%). The population sizes obtained by a manual gating (see Figure 3) are considered ground-truth as the 3 clusters are separable for these controlled experiments. By applying the same procedure as in Section IV-A, the minimum ranks (and their associated runtimes) that provide the 3 cell populations were found for each method with $I = 20$. Table II shows that SQUAREM-PMF identifies the MP cluster with a lower rank compared with CTF3D and PCTF3D. Moreover, SQUAREM-PMF gives a more accurate estimate of $\widehat{\lambda}_{\text{MP}}$, but at the cost of a higher computational time.

VI. Conclusions and Discussion

In this paper, three tensor-based probability mass function estimation methods for flow cytometry data analysis were presented: SQUAREM-PMF, CTF3D and PCTF3D. For all three methods, the distribution is modelled with a naïve Bayes model whose complexity

TABLE I
PROPERTIES OF THE 3 POPULATIONS USED IN THE CONTROLLED EXPERIMENT. + IS HIGH MARKER EXPRESSION AND - LOW EXPRESSION.

	Marker expression			
Population	CFSE	CD4	CTV	MHCII
Macrophage	-	-	+	+
Lymphocyte B	+	-	-	++
Lymphocyte T	-	+-	-	-

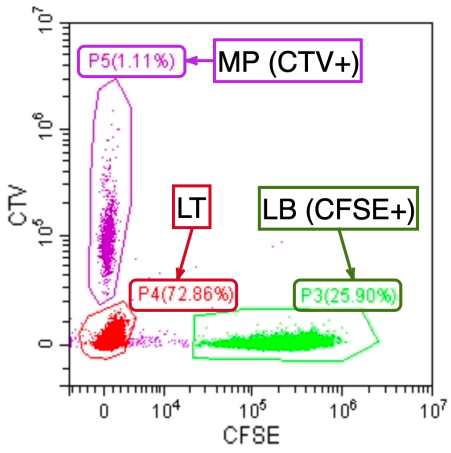

Fig. 3. Manual gating of the 3 populations. The plot shows three gates : P3 groups CFSE+ cells (LB), P4 groups LT cells and P5 CTV+ cells (MP). *Logicle* scale is used on this plot [15].

remains linear with the number of variables. However, the three presented approaches have different complexities, as each method estimates canonical polyadic decomposition factors differently. To compare the approaches, a sensitivity study was performed showing that, compared to the approaches based on marginals, SQUAREM-PMF is able to identify small clusters at lower ranks . Experiments on a real flow cytometry dataset corroborate this conclusion. However, the runtime of SQUAREM-PMF depends on the number of samples. According to the results of our experiments, for large numbers of data points (typical for flow cytometry application) CTF3D and PCTF3D have much lower computational time, but can achieve a comparable accuracy to SQUAREM-PMF. We believe that, for large datasets, a combination of marginal- and ML-based approaches needs to be developed, which we leave as a future research question.

TABLE II
SENSITIVITY EXPERIMENT FOR $N = 4$ CONTROLLED EXPERIMENTS.

Gating	CTF3D	PCTF3D	SQUAREM-PMF
$\lambda_{\text{MP}} = 20.7\%$	19.9% $R = 4$ 1.2s	19.3% $R = 4$ 1s	20.5% $R = 4$ 5.3s
$\lambda_{\text{MP}} = 8\%$	6% $R = 11$ 4.5s	5.3% $R = 15$ 15s	7.9% $R = 7$ 31s
$\lambda_{\text{MP}} = 1.1\%$	0.71% $R = 29$ 35s	0.76% $R = 31$ 25s	1.4% $R = 11$ 170s

REFERENCES

[1] S. P. Perfetto, P. K. Chattopadhyay, and M. Roederer, "Seventeen-colour flow cytometry: unravelling the immune system," *Nature Reviews Immunology*, vol. 4, no. 8, pp. 648–655, 2004.

[2] P. Qiu, E. F. Simonds, S. C. Bendall, K. D. Gibbs Jr, R. V. Bruggner, M. D. Linderman, K. Sachs, G. P. Nolan, and S. K. Plevritis, "Extracting a cellular hierarchy from high-dimensional cytometry data with SPADE," *Nature biotechnology*, vol. 29, no. 10, pp. 886–891, 2011.

[3] E. D. Amir, K. L. Davis, M. D. Tadmor, E. F. Simonds, J. H. Levine, S. C. Bendall, D. K. Shenfeld, S. Krishnaswamy, G. P. Nolan, and D. Pe'er, "viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia," *Nature biotechnology*, vol. 31, no. 6, pp. 545–552, 2013.

[4] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," *Journal of Mathematics and Physics*, vol. 6, no. 1-4, pp. 164–189, 1927.

[5] N. Kargas, N. D. Sidiropoulos, and X. Fu, "Tensors, learning, and "Kolmogorov extension" for finite-alphabet random vectors," *IEEE Transactions on Signal Processing*, vol. 66, no. 18, pp. 4854–4868, 2018.

[6] P. Flores, G. Harlé, A. Notarantonio, K. Usevich, M. d'Aveni, S. Grandemange, M. Rubio, and D. Brie, "Coupled tensor factorization for flow cytometry data analysis," in *Proc. IEEE 32nd International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2022.

[7] A. Yeredor and M. Haardt, "Maximum Likelihood Estimation of a Low-Rank Probability Mass Tensor From Partial Observations," *IEEE Signal Processing Letters*, vol. 26, no. 10, pp. 1551–1555, Oct. 2019.

[8] J. K. Chege, M. J. Grasis, A. Manina, A. Yeredor, and M. Haardt, "Efficient Probability Mass Function Estimation from Partially Observed Data," in *Proc. IEEE 56th Asilomar Conference on Signals, Systems, and Computers*, Oct 2022.

[9] A. Yeredor and M. Haardt, "Estimation of a Low-Rank Probability-Tensor from Sample Sub-Tensors via Joint Factorization Minimizing the Kullback-Leibler Divergence," in *Proc. IEEE 27th European Signal Processing Conference (EUSIPCO)*, Sept. 2019.

[10] N. Kargas and N. D. Sidiropoulos, "Learning mixtures of smooth product distributions: Identifiability and algorithm," in *Proc. 22nd International Conference on Artificial Intelligence and Statistics*, Apr 2019.

[11] T. Kolda and B. Bader, "Tensor Decompositions and Applications," *SIAM Review*, vol. 51, pp. 455–500, Aug. 2009.

[12] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al., "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[13] K. Huang, N. D. Sidiropoulos, and A. P. Liavas, "A Flexible and Efficient Algorithmic Framework for Constrained Matrix and Tensor Factorization," *IEEE Transactions on Signal Processing*, vol. 64, no. 19, pp. 5052–5065, Oct. 2016.

[14] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup, "Scalable tensor factorizations for incomplete data," *Chemometrics and Intelligent Laboratory Systems*, vol. 106, no. 1, pp. 41–56, 2011.

[15] D. R. Parks, M. Roederer, and W. A. Moore, "A new "Logicle" display method avoids deceptive effects of logarithmic scaling for low signals and compensated data," *Cytometry Part A: The Journal of the International Society for Analytical Cytology*, vol. 69, no. 6, pp. 541–551, 2006.