

CytOpT: Optimal Transport with Domain Adaptation for Interpreting Flow Cytometry data

Paul Freulon, Jérémie Bigot, Boris Hejblum

▶ **To cite this version:**

Paul Freulon, Jérémie Bigot, Boris Hejblum. CytOpT: Optimal Transport with Domain Adaptation for Interpreting Flow Cytometry data. Annals of Applied Statistics, 2023, 17 (2), pp.1086-1104. 10.1214/22-AOAS1660 . hal-03100405

HAL Id: hal-03100405
https://inria.hal.science/hal-03100405

Submitted on 7 Jan 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

arXiv:2006.09003v3 [stat.AP] 18 Jun 2020

CytOpT: Optimal Transport with Domain Adaptation for Interpreting Flow Cytometry data

Paul Freulon *[1,2], Jérémie Bigot[1,2], and Boris P. Hejblum[1,3,4]

[1]Université de Bordeaux, Bordeaux, 33000, France.
[2]Institut de Mathématiques de Bordeaux et CNRS (UMR 5251), 33400 Talence, France.
[3]Bordeaux Population Health Research Center Inserm U1219, Inria SISTM, 33000 Bordeaux, France.
[4]Vaccine Research Institute (VRI), 94010 Créteil, France.

Abstract

The automated analysis of flow cytometry measurements is an active research field. We introduce a new algorithm, referred to as `CytOpT`, using regularized optimal transport to directly estimate the different cell population proportions from a biological sample characterized with flow cytometry measurements. We rely on the regularized Wasserstein metric to compare cytometry measurements from different samples, thus accounting for possible mis-alignment of a given cell population across samples (due to technical variability from the technology of measurements). In this work, we rely on a supervised learning technique based on the Wasserstein metric that is used to estimate an optimal re-weighting of class proportions in a mixture model from a source distribution (with known segmentation into cell sub-populations) to fit a target distribution with unknown segmentation. Due to the high-dimensionality of flow cytometry data, we use stochastic algorithms to approximate the regularized Wasserstein metric to solve the optimization problem involved in the estimation of optimal weights representing the cell population proportions in the target distribution. Several flow cytometry data sets are used to illustrate the performances of `CytOpT` that are also compared to those of existing algorithms for automatic gating based on supervised learning.

Keywords: Automatic gating; flow cytometry; Optimal Transport; Stochastic Optimization

1 Introduction

Flow cytometry is a high-throughput biotechnology used to characterize a large amount of cells from a biological sample. Flow cytometry is paramount to many biological and immunological research with applications (for instance) in the monitoring of the immune system of HIV patients by counting the number of CD4 cells.

The first step to characterize cells from a biological sample with flow cytometry is to stain those cells. Specifically, cells are stained with multiple fluorescently-conjugated monoclonal antibodies directed to the cellular markers of interest. Then, the cells flow one by one through the cytometer laser beam. The scattered light is characteristic to the biological markers of the cells N.Aghaeepour et al. [2013]. Thus, from a biological sample analyzed by a flow cytometer, we get a data set $X_1, ..., X_I$ where each observation X_i corresponds to a single cell crossing the laser beam. For an observation $X_i \in \mathbb{R}^d$, the coordinate $X_i^{(m)}$ corresponds to the light intensity emitted by the fluorescent antibody attached to the biological marker m. Interestingly, such a data set may also be considered as a discrete probability distribution $\frac{1}{I}\sum_{i=1}^{I} \delta_{X_i}$ with support in $\mathbb{R}^d$, this is the point of view taken in this paper.

One can thus assess individual characteristics at a cellular level for samples composed of hundreds of thousands of cells. The development of this technology now leads to state-of-the-art cytometers that can measure up to 50 biological markers at once on a single cell Y.Saeys et al. [2016]. The constant increase of the number of measured markers allows to identify more precisely the different cell populations and subtypes present in the biological sample.

*correspondence: paul.freulon@math.u-bordeaux.fr

The analysis of cytometry data is generally done manually, by drawing geometric shapes (referred to as "gates") around populations of interest in a sequence of two-dimensional data projections. Such manual analysis features several drawbacks: i) it is extremely time-consuming ; ii) manual gating lacks reproducibility across different operators N.Aghaeepour et al. [2013]. To overcome these shortcomings, several automated methods have been proposed N.Aghaeepour et al. [2013]. Those automated approaches aim at a clustering of the flow cytometry data to derive the proportions of the cell populations that are in the biological sample. Some methods follow an unsupervised approach. For instance `Cytometree` D.Commenges et al. [2018] is an algorithm based on the construction of a binary tree. Other unsupervised methods which perform model based clustering have been proposed, see e.g. Y.Ge and S.C.Sealfon [2012], B.P.Hejblum et al. [2019] To improve the accuracy of the classification, supervised machine learning techniques have been applied to flow cytometry data analysis. Among those techniques, one may cite `DeepCyTOF` introduced in H.Li et al. [2017] which is based on deep-learning algorithms to gate cytometry data. In M.Lux et al. [2018], the authors introduced `flowlearn`, a method that uses manually gated samples to predict gates on other samples. We also want to mention a new supervised approach named `OptimalFlow` developed by Barrio et al. [2019]. This method drew our attention as `OptimalFlow` introduces the Wasserstein distance to quantify the discrepancy between cytometry data sets.

In spite of the numerous efforts to automate cytometry data analysis, manual gating remains the gold-standard for benchmarking. The fact that automated analysis has not outdone manual gating can be explained, at least in part, by the significant variability of flow cytometry data. The variability of the flow cytometry is twofold. First, variability is induced by biological heterogeneity across the samples analyzed. For instance, we cannot expect the same cell subtypes proportions within the biological sample of a healthy patient and the biological sample of a sick patient. In addition to this variability of the biological phenomena of interest, there is a technical variability. This variability appears during the process of flow cytometry analysis. For instance, differences in the staining procedure, in the data acquisition settings or cytometers performances are very likely to happen and lead to undesirable variability between flow cytometry data. This variability of flow cytometry data makes the development of automated methods a challenging task.

1.1 An illustrative data set

As an illustrative example, we shall analyze in this paper the flow cytometry data from the T-cell panel of the Human Immunology Project Consortium (HIPC) – publicly available on ImmuneSpace R.Gottardo et al. [2014]. Seven laboratories stained three replicates (denoted A, B, and C) of three cryo-preserved biological samples denoted patient 1, 2, and 3 (e.g. cytometry measurements from the Stanford laboratory for replicate C from patient 1 will be denoted as the "Stanford1C" data set). After performing cytometry measurements in each center, the resulting FCS files were manually gated centrally for quantifying 10 cell populations: CD4 Effector (CD4 E), CD4 Naive (CD4 N), CD4 Central memory (CD4 CM), CD4 Effector memory (CD4 EM), CD4 Activated (CD4 A), CD8 Effector (CD8 E), CD8 Naive (CD8 N), CD8 Central memory (CD8 CM), CD8 Effector memory (CD8 EM) and CD8 Activated (CD8 A). Hence, for these data sets, a manual clustering is at our disposal to evaluate the performances of automatic gating methods. Those 10 sub populations have a size that range from 15 554 observations for the smallest data set to 112 318 observations for the largest data set. For each cell we have the measurement of seven markers that thus lead to cytometry observations X_i that belong to $\mathbb{R}^d$ with $d = 7$. A 3D projection using three markers is displayed in Figure 1 for the "Stanford1A" data set with the corresponding manual gating into 10 clusters.

1.2 Main contributions

We propose a new supervised method to estimate the relative proportions of the different cell sub-types in a biological sample analyzed by a cytometer. Our approach aims at finding an optimal re-weighting of class proportions in a mixture model between a source data set (with known segmentation into cell sub-populations) to fit a target data set with unknown segmentation. This estimation of the class proportions is done without any preliminary clustering of the target cytometry data. To the best of our knowledge, all the previous automated methods are based on the classification of the cells of the biological sample to deduce a class proportions estimation. However, from a clinical perspective, the relevant information is the class proportions, while the clustering of individual cells is simply a mean to an end to get there. Our method can nonetheless be extended in order to obtain a clustering of cytometry data.

We believe that going straight to the estimation of class proportions is an original approach to tackle

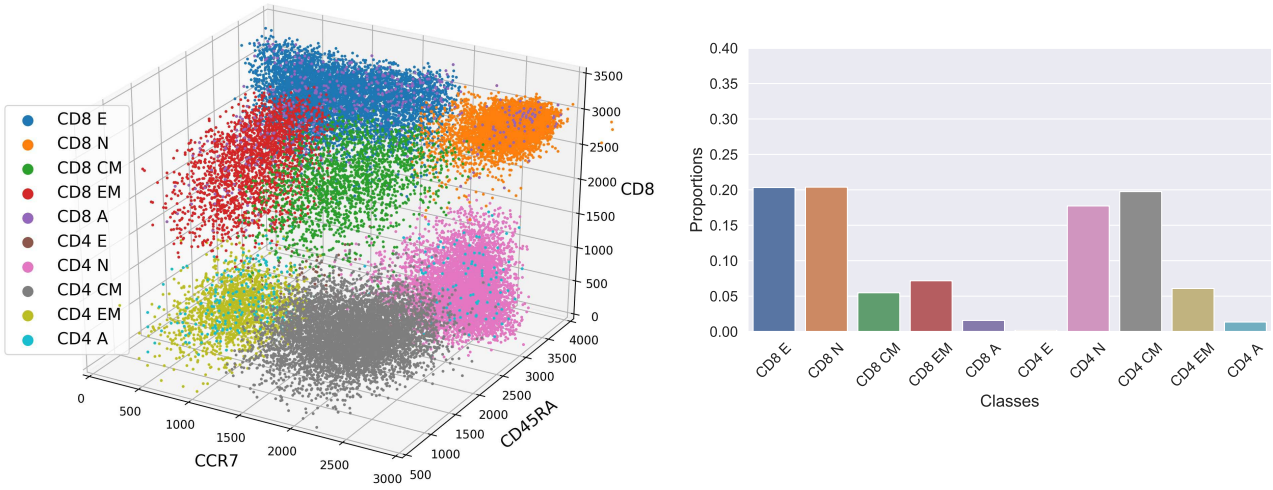


Figure 1: **Example flow cytometry data set of a biological sample from Stanford patient 1 replicate A.** Left: Manual clustering of the cytometry data. Existing automated methods target a clustering in order to derive the class proportions. Right: Class proportion derived from the manual gating. Our method `CytOpT` aims at going straight to the estimation of class proportions without clustering the cytometry data.

the issue of flow cytometry data analysis. Hence, the result of our algorithm is not a clustering of the cells analyzed by the cytometers but a vector $h = (h_1, ..., h_K)$, where each coefficient h_k amounts to the percentage of the k^{th} population cell among all the cells analyzed in the target sample. To do so, we rely on tools derived from optimal transport and regularized Wasserstein distance between probability measures.

Optimal transport has recently gained interest in machine learning and statistics. Indeed, the introduction of approximate solvers that can tackle large dimension problems allowed to move beyond the high computational cost of optimal transport. Thus, optimal transport has found various applications in machine learning for regression H.Janati et al. [2018], classification R.Flamary et al. [2018] and generative modeling M.Arjovsky et al. [2017]. Those computational progress in optimal transport have also allowed its use in imaging sciences J.Solomon et al. [2015]. Recent progress in applied optimal transport has been fueled by the development of efficient, large-scale optimization algorithms for problems in this domain. In particular, our method rely on stochastic algorithms to handle the computational cost of optimal transport.

Our approach involves the comparison of a source data set whose gating into sub-population cells of interest is known with a target data set whose gating is unknown. We estimate the proportion of each sub-population cell in the target data by minimizing the regularized Wasserstein distance between the target distribution and a re-weighted vector of class proportions in the source distribution. Since the Wasserstein distance is able to capture the underlying geometry of the measurements, it has the ability to make meaningful comparisons between distributions whose supports do not overlap and to quantify potential spatial shifts. Our work demonstrates the benefits of using Wasserstein distance for the analysis of the cytometry data as it allows to handle the technical variability induced by different settings of measurements.

1.3 Organization of the paper

In Section 2 we present the Wasserstein distance and its regularized version. We also introduce the stochastic algorithm that is used to compute the regularized Wasserstein distance. Section 3 details our estimation method that yields an estimate of the class proportions in an unsegmented data set. Finally, in Section 4 we demonstrate the performance of `CytOpT` on the T-cell panel proposed by the Human Immunology Project Consortium (HIPC) described previously.

2 Mathematical framework

Optimal transport allows the definition of a metric between two probability distributions α and β supported on $\mathbb{R}^d$. This metric is informally defined as the lowest cost to move the mass from one probability measure, the source measure α, onto the other, the target measure β.

As optimal transport handles probability distributions, we must describe a cytometry data set $X_1, ..., X_I$ where each X_i belongs to $\mathbb{R}^d$ as a probability distribution in $\mathbb{R}^d$. A natural choice to move from observations to probability measure is to use the empirical measure. Therefore, for a data set $X_1, ..., X_I$, we consider its empirical measure defined as $\frac{1}{I}\sum_{i=1}^{I}\delta_{X_i}$. In this work, we shall consider optimal transport between discrete measures on $\mathbb{R}^d$, and we denote by $\alpha = \sum_{i=1}^{I} a_i\delta_{X_i}$ and $\beta = \sum_{j=1}^{J} b_j\delta_{Y_j}$ two such measures. Note that a_i is not necessarily equal to $1/I$ as a re-weighting of the source measure will be considered. Optimal transport then seeks a plan that moves the probability measure α to the probability measure β with a minimum cost. The transportation cost is encoded by a function $c : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$ where $c(x, y)$ represents the cost to move one unit of mass from x to y. In the discrete setting the transportation cost is encoded by a matrix $C \in \mathbb{R}^{I\times J}$ where $C_{i,j} = c(X_i, Y_j)$.

By denoting $\Pi(\alpha, \beta) = \{P \in \mathbb{R}^{I\times J} : P1_J = a \text{ and } P^T 1_I = b\}$ all the coupling matrices from α to β, minimizing the transportation cost from α to β boils down to the following optimization problem:

$$W_c(\alpha, \beta) = \inf_{\pi\in\Pi(\alpha,\beta)} \sum_{i=1}^{I}\sum_{j=1}^{J} C_{i,j}P_{i,j} = \inf_{\pi\in\Pi(\alpha,\beta)} \langle C, P\rangle \tag{1}$$

If the cost matrix c is defined through a distance on $\mathbb{R}^d$, for instance if $c(x, y) = ||x - y||_p^p$, then the quantity $W_p(\alpha, \beta) = (W_c(\alpha, \beta))^{1/p}$ defines a distance between probability measures on $\mathbb{R}^d$. All along this work the coefficient $C_{i,j}$ of the cost matrix is defined as the squared Euclidean distance between x_i and y_j, i.e. $C_{i,j} = ||x_i - y_j||_2^2$, and we use the notation $W(\alpha, \beta) = W_2^2(\alpha, \beta)$. However, the cost to evaluate the Wasserstein distance between two discrete probability distributions with support of equal size N is generally of order $O(N^3 \log(N))$. To allow the evaluation of the Wasserstein distance between large data sets M.Cuturi [2013] has proposed to add an entropic regularization term to the linear optimization problem (1) in order to accelerate the computation of an approximate solution. The regularized problem is

$$W^\varepsilon(\alpha, \beta) = \inf_{\pi\in\Pi(\alpha,\beta)} \langle C, P\rangle + \varepsilon H(P), \tag{2}$$

where $H : \mathbb{R}_+^{I\times J} \to \mathbb{R}$ is defined for $P \in \mathbb{R}_+^{I\times J}$ by $H(P) = \sum_{i,j} (\log(P_{i,j}) - 1) P_{i,j}$. The entropic regularization of the Kantorovich problem (2) leads to an approximation of the Wasserstein distance which can be calculated in $O(N^2 \log(N))$ operations if the two distributions have a support of size N G.Peyré et al. [2019].

2.1 Calculation of regularized Wasserstein distances

In this work, the regularized Wasserstein distance is calculated with a statistical procedure based on the Robbins-Monro algorithm for stochastic optimization. This way to calculate the Wasserstein distance is investigated in A.Genevay et al. [2016] and B.Bercu and J.Bigot [2020]. The keystone of this approach is that the regularized Wasserstein problem can be written as the following stochastic optimization problem:

$$W^\varepsilon(\alpha, \beta) = \max_{u\in\mathbb{R}^I} \mathbb{E}[g_\varepsilon(Y, u)], \tag{3}$$

where Y is a random variable with distribution β, and g_ε is defined as:

$$g_\varepsilon(y, u) = \sum_{i=1}^{I} u_i a_i + u_{c,\varepsilon}(y) - \varepsilon, \tag{4}$$

where $u_{c,\varepsilon}(y_j) = \varepsilon\left(\log(b_j) - \log\left(\sum_{i=1}^{I}\exp\left(\frac{u_i - c(x_i,y_j)}{\varepsilon}\right)\right)\right)$. We point out that the solution P_ε of the primal problem (2) is recovered from any $u^* \in \mathbb{R}^I$ solution of the semi-dual problem (3) as:

$$(P_\varepsilon)_{i,j} = \exp\left(\frac{u_i^* + u_{c,\varepsilon}^*(y_j) - c(x_i, y_j)}{\varepsilon}\right) \tag{5}$$

Formulation (3) and the fact that for all $y \in \mathbb{R}^d$, the function $g(y,.)$ is concave lead us to estimate the vector u^* by the Robbins-Monro algorithm H.Robbins and S.Monro [1951] given, for all $n \geq 0$, by:

$$\widehat{U}_{n+1} = \widehat{U}_n + \gamma_{n+1} \nabla_u g_\varepsilon(Y_{n+1}, \widehat{U}_n), \tag{6}$$

where the initial value $\hat{U}_0$ is a random vector which can be arbitrarily chosen, $Y_1, ..., Y_{n+1}$ is an i.i.d. sequence of random variables sampled from the distribution β, and $(\gamma_n)_{n\geq 0}$ is a positive sequence of real numbers decreasing toward zero satisfying

$$\sum_{n=1}^{\infty} \gamma_n = +\infty \qquad \text{and} \qquad \sum_{n=1}^{\infty} \gamma_n^2 < +\infty. \tag{7}$$

It follows from M.Cuturi [2013] that the maximizer u^* of (3) is unique up to a scalar translation of the form $u^* - t1_I$ for any $t \in \mathbb{R}$. Throughout this paper, we shall denote by u^* the maximizer of (3) satisfying $\langle u^*, 1_I \rangle = 0$ which means that u^* belongs to $\langle 1_J \rangle^\perp$ where $\langle 1_J \rangle$ is the one-dimensional subspace of $\mathbb{R}^I$ spanned by 1_J. As shown in B.Bercu and J.Bigot [2020], to obtain a consistent estimator of u^*, one must slightly modify the Robbins-Monro algorithm by requiring that $\widehat{U}_0$ belongs to $\langle 1_I \rangle^\perp$. To approximate the regularized Wasserstein distance $W^\varepsilon(\alpha, \beta)$ B.Bercu and J.Bigot [2020] proposed the recursive estimator:

$$\widehat{W}_n = \frac{1}{n} \sum_{k=1}^{n} g_\varepsilon(Y_k, \widehat{U}_{k-1}). \tag{8}$$

and throughout this work, we will use this estimator.

2.2 Statistical model

Let us consider $X_1^s, ..., X_I^s$ the cytometry measurement from a first biological sample. All along our work, this sample will be referred to as the source sample or the source observations. The distribution of the source sample is modeled by a mixture of K distributions: $\alpha = \sum_{k=1}^K \rho_k \alpha_k$, where each term of the mixture corresponds to the cytometry measurements of one type of cell. For instance, α_k represents the underlying distribution behind the cytometry measurements of the CD4 effector T cells that are in the biological sample. The weights $\rho \in \Sigma_K$ are coefficients lying in the probability simplex $\Sigma_K = \{h \in \mathbb{R}_+^K \ : \ \sum_{k=1}^K h_k = 1\}$, and ρ_k represents the proportion of one cell sub-type among all the cells found in the sample.

We also consider a second set of cytometry measurements $X_1^t, ..., X_J^t$. This set typically corresponds to cytometry measurements from a second biological sample that may come from an other patient and the cytometry analysis may have been performed in a different laboratory. All along our work, this sample will be referred to as the target sample or the target observations. We assume that the underlying distribution of the the target observations is another mixture of K distributions: $\beta = \sum_{k=1}^K \pi_k \beta_k$, where $\pi \in \Sigma_K$ represents the unknown class proportions and β_k corresponds to the distribution of the k^{th} cell population. Note that, in the framework of flow cytometry, we cannot make the assumption that $\alpha_k = \beta_k$ neither that $\rho_k = \pi_k$. Indeed, ρ could differ from π due to biological differences. For instance if the source sample comes from a healthy patient and the target sample comes from a sick patient, it is likely that π will feature significant different from ρ. Moreover, between each component β_k and α_k there could be a difference (e.g. location shift) due to the technical variability of the cytometry measurement.

These potential differences between the distributions of two flow cytometry measurements make the development of supervised methods to infer the unknown proportions π in the target data from those of the source data a difficult task. In this work, we rely on the geometric properties of the Wasserstein distance to handle the differences between samples that are due to technical reasons or inter-variability between healthy and sick patients. In the sequel, we present a supervised algorithm that estimates the class proportions π in the target distribution. In this method, we only use the segmentation of one source sample $X_1^s, ..., X_I^s$ to estimate the class proportions in an unsegmented target sample $X_1^t, ..., X_J^t$.

3 Class proportion estimation

Recall that we are in the following learning framework. From a cytometry data set $X_1^s, ..., X_I^s$ which is segmented in K classes $C_1, ..., C_K$ we aim at estimating the class proportions in an other data set $X_1^t, ..., X_J^t$ for which a clustering is not available. While state-of-the-art automated methods in cytometry

data analysis attempt to classify the observations $X_1^t, ..., X_J^t$, the method that we propose will go straight to the estimation of the class proportions in the unsegmented target data set . To do so, we borrow ideas from the domain adaptation technique proposed in I.Redko et al. [2018] where it is proposed to re-weight the source observations by searching for weights minimizing the Wasserstein distance between the re-weighted source measure and the target measure. Contrary to I.Redko et al. [2018], we handle the Wasserstein distance and the minimization problem by a stochastic approach. Indeed the large number of observations produced by flow cytometry makes stochastic techniques competitive to handle the high-dimensionality of such data sets.

3.1 A new estimator of the class proportions

We now specify the definition of the estimator of the class proportions in the target data set. For the target sample $X_1^t, ..., X_J^t$, the segmentation into various cell sub-types is not available, hence we define the empirical target measure $\hat{\beta}$ as $\hat{\beta} = \frac{1}{J}\sum_{j=1}^{J} \delta_{X_j^t}$. From the source observations $X_1^s, ..., X_I^s$, we define the empirical source measure $\hat{\alpha}$ as $\hat{\alpha} = \frac{1}{I}\sum_{i=1}^{I} \delta_{X_i^s}$

Then, the knowledge of the gating of the source data allows us to re-write the measure $\hat{\alpha}$ as a mixture of probability measures where each component corresponds to a known sub-population of cells in the data

$$\hat{\alpha} = \sum_{k=1}^{K} \frac{n_k}{I} \left(\sum_{i:X_i^s \in C_k} \frac{1}{n_k} \delta_{X_i^s} \right) = \sum_{k=1}^{K} \frac{n_k}{I} \hat{\alpha}_k, \tag{9}$$

where $n_k = \#C_k$ and $\hat{\alpha}_k = \sum_{i:X_i^s \in C_k} \frac{1}{n_k} \delta_{X_i^s}$. Namely, the component $\hat{\alpha}_k$ is the empirical measure of the observations that belong to the sub-population C_k (known class).

Then, instead of only considering the true class proportions $(n_1/I, ..., n_K/I)$ in the source data set, we can re-weight the clusters C_k in the empirical distribution as desired. Indeed, for a probability vector $h = (h_1, ..., h_K) \in \Sigma_K$ we can define the measure $\hat{\alpha(h)}$ that corresponds to the re-weighted measure $\hat{\alpha}$ such that for all $k \in \{1, ..., K\}$ the component $\hat{\alpha}_k$ amounts for h_k in the measure $\hat{\alpha}(h)$. In mathematical terms, the measure $\hat{\alpha}(h)$ is thus defined by:

$$\hat{\alpha}(h) = \sum_{k=1}^{K} h_k \hat{\alpha}_k \tag{10}$$

Our domain adaptation technique to derive the class proportions in the target data is based on the re-weighting of the source data in order to minimize the Wasserstein regularized distance (2) between the re-weighted source empirical distribution (10) and the target empirical distribution. The main idea is that the source distribution will get closer to the target distribution as the class proportions in its re-weigthed version get closer to the class proportions of the target distribution. Thus, we propose to estimate the weights $\pi = (\pi_1, ..., \pi_K) \in \Sigma_K$ of the underlying distribution α_t behind the observations $X_1^t, ..., X_J^t$ of the unlabelled target data set by

$$\hat{\pi} \in \underset{h \in \Sigma_K}{\arg\min}\, W^{\varepsilon}(\hat{\alpha}(h), \hat{\beta}) \tag{11}$$

The combination of the estimator $\hat{\pi}$ and the algorithms described in Subsection 3.4 to solve the associated minimization problem (11) will be refered to as `CytOpT`.

3.2 Extension of `CytOpT` to a soft assignment method

While our method aims at estimating the class proportions and not to classify the target observations, regularized optimal transport offers a natural soft assignment method which can be used to derive a soft classification of the target data set, as illustrated in Figure 5. Assuming that we have access to the optimal transport plan P_ε with respect to the regularized problem (2), the coefficient $(P_\varepsilon)_{i,j}/b_j$ can be interpreted as the probability that X_i^s is assigned to X_j^t. Here, b_j is the weight associated to the observation X_j^t. Thus, the probability $\gamma_j^{(k)}$ that X_j^t belongs to the class C_k is $\gamma_j^{(k)} = \frac{1}{b_j}\sum_{i=1}^{I} 1_{X_i^s \in C_k} (P_\varepsilon)_{i,j}$ By choosing the class with highest probability we can derive a classification for the observation X_j^t. One can compute an approximate $\hat{P}$ of P_ε by plugging $\widehat{U}$ the Robbins-Monro approximation of the optimal dual vector u^* in formula (5).

Thus, we can derive an automatic gating of the target data thanks to optimal transport. However, the transfer of the classification from the source data set toward the target data set requires to compute all the columns of the optimal transport plan. Therefore, obtaining a classification in the target data call for additional calculations. Nevertheless, in flow cytometry, the information that matters from a clinical point of view is the relative proportions of the different cell sub-types. Hence, we focus on the estimation of class proportions in this work.

3.3 Measure of performance

To evaluate our approach, the benchmark class proportions $\pi = (\pi_1, ..., \pi_K) \in \Sigma_K$, will be defined thanks to the manual segmentation of the target observations. We recall that our algorithm does not make use of the segmentation of the target data set, and that it is only used to evaluate our method.

In flow cytometry data analysis the F-measure is a popular tool to asses the performance of the clustering methods. As our method does not yields a clustering but an estimate $\hat{\pi}$ of the class proportions π, we cannot rely on this measure. Indeed, `CytOpT` yields a probability vector $\hat{\pi} \in \Sigma_K$ that we wish to be the closest to the class proportion $\pi \in \Sigma_K$ in the target data set. Hence, a natural way to measure the discrepancy between $\hat{\pi}$ and π, is the Kullback-Leilbler divergence. Thus, with an estimation $\hat{\pi} = (\hat{\pi}_1, ..., \hat{\pi}_K)$ and a benchmark $\pi = (\pi_1, ..., \pi_K)$, to asses the quality of the estimator $\hat{\pi}$, we compute the Kullback-Leibler divergence KL defined as $\mathrm{KL}(\hat{\pi}|\pi) = \sum_{k=1}^{K} \hat{\pi}_k \log\left(\frac{\hat{\pi}_k}{\pi_k}\right)$.

3.4 A brief overview of the minimization procedure

The optimization problem (11) leading to the estimator $\hat{\pi}$ of the class proportions does not have a solution in a closed form expression, and a numerical approximation is needed. We propose two methods to solve this optimization problem. In this section, we only give the mains ideas of these two methods without focusing on the technical details. For a more thorough presentation, we refer the reader to Appendix A.

3.4.1 Descent-Ascent procedure

First, we re-write problem (11) as

$$\min_{h \in \Sigma_K} W^{\varepsilon}(\hat{\alpha}(h), \hat{\beta}) = \min_{h \in \Sigma_K} \max_{u \in \mathbb{R}^I} \mathbb{E}[g_{\varepsilon}(Y, u, h)], \tag{12}$$

where Y is a random variable with distribution $\hat{\beta}$, and g_{ε} is the function defined by (4) that can be easily computed.

Second, to avoid projecting h on the simplex Σ_K at each iteration, we re-parameterize problem (12) with a soft max function σ whose complete expression is given in (19). Hence, our new minimization problem is

$$\min_{z \in \mathbb{R}^K} W^{\varepsilon}(\hat{\alpha}(\sigma(z)), \hat{\beta}) = \min_{z \in \mathbb{R}^K} \max_{u \in \mathbb{R}^I} \mathbb{E}[g_{\varepsilon}(Y, u, \sigma(z))] \tag{13}$$

The gradient of the function $F : z \mapsto W^{\varepsilon}(\hat{\alpha}(\sigma(z)), \hat{\beta})$ can be expressed for $z \in \mathbb{R}^K$ as $\nabla_z F(z) = (\Gamma J_{\sigma}(z))^T u_z^*$ where u_z^* is a maximizer of the function $u \mapsto \mathbb{E}[g_{\varepsilon}(Y, u, \sigma(z))]$. The expressions of Γ and J_{σ} can be found in (15) and (22) respectively.

Hence, a natural way to solve (13) is to use a gradient descent ascent algorithm where we alternate between a gradient descent on the variable z and a gradient ascent on the variable u. The approximate of the variable u_z^* allows us to compute the gradient in the outer loop. An interesting feature of our algorithm is that to compute an approximate U_z of the vector u_z^* we rely on the Robbins-Monro algorithm (6). Then, by computing $\hat{\omega}(z) = (\Gamma J_{\sigma}(z))^T U_z$ we get a stochastic approximation of $\nabla_z F(z)$ to proceed with the gradient ascent.

3.4.2 Minmax swapping procedure

To avoid the use of an inner-loop algorithm, we borrow ideas developed in M.Ballu et al. [2020]. By adding an entropic penalization φ defined in (27) to problem (11), we obtain the following optimization problem:

$$\min_{h \in \Sigma_k} W^{\varepsilon}(\hat{\alpha}(h), \hat{\beta}) + \lambda \varphi(h) = \min_{h \in \Sigma_k} \max_{u \in \mathbb{R}^I} \mathbb{E}[g_{\varepsilon}(Y, u, h)] + \lambda \varphi(h) \tag{14}$$

Swapping the min and the max, and using the fact that, for a given $u \in \mathbb{R}^I$, we can compute an explicit solution $h(u) \in \Sigma_K$ of the problem $\min_{h \in \Sigma_K} \mathbb{E}[g_{\varepsilon}(Y, u, h)] + \lambda \varphi(h)$, such that, for $k \in \{1, ..., K\}$, $(h(u))_k =$

Algorithm 1: Solving $\min_{z\in\mathbb{R}^K}\max_{u\in\mathbb{R}^I}\mathbb{E}_{Y\sim\hat{\beta}}[g_\varepsilon(Y,u,\sigma(z))]$

$z \leftarrow 1_K$
for $l \leftarrow 1$ **to** n_{out} **do**
  $U \leftarrow$ an arbitrary vector in $\mathbb{R}^I$
  for $k \leftarrow 1$ **to** n_{in} **do**
    $Y \sim \hat{\beta}$
    $U \leftarrow U + \gamma_k \nabla_u g_\varepsilon(Y, U, \sigma(z))$
  end
  `/* Approximation of the gradient of` $z \mapsto W^\varepsilon(\hat{\alpha}(\sigma(z)), \hat{\beta})$ `*/`
  $\hat{\omega}(z) \leftarrow (\Gamma J_\sigma(z))^T U$
  $z \leftarrow z - \eta\hat{\omega}(z)$
end
`/* Computation of the estimator of the class proportion` $\hat{\pi}$ `*/`
return $\hat{\pi} = \sigma(z)$

$\frac{\exp\left(-\frac{(\Gamma^T u)_k}{\lambda}\right)}{\sum_{l=1}^K \exp\left(-\frac{(\Gamma^T u)_l}{\lambda}\right)}$, where Γ the linear operator defined in (15). Thus, adding an entropy penalization, induces a natural parameterization of the simplex by the soft-max function. Then, it follows from this result that problem (14) boils down to $\max_{u\in\mathbb{R}^I}\mathbb{E}_{Y\sim\hat{\beta}}[f_{\varepsilon,\lambda}(Y,u)]$, where the complete expression of $f_{\varepsilon,\lambda}$ can be found in (34). Once again, we solve this problem using a Robbins-Monro algorithm, similarly with (6).

Algorithm 2: Solving $\max_{u\in\mathbb{R}^I}\mathbb{E}_{Y\sim\hat{\beta}}[f_{\varepsilon,\lambda}(Y,u)]$

$U \leftarrow 0_I$
for $l \leftarrow 1$ **to** n **do**
  $Y \sim \hat{\beta}$
  $U \leftarrow U + \gamma_l \nabla_u f_{\varepsilon,\lambda}(Y, U)$
end
`/* Computation of the estimator of the class proportion` $\hat{\pi}$ `*/`
$\hat{\pi} \leftarrow h(U)$
return $\hat{\pi}$

3.4.3 Comparison of the minimization procedures

As said in Subsection 3.3, we asses the performance of our estimator $\hat{\pi}$ by computing the Kullback-Leibler divergence between $\hat{\pi}$ and π the benchmark class proportions. Figure 2 displays the evolution of the Kullback-Leibler divergence along the iterations of the two minimization procedures. For the descent-ascent procedure, the iterations displayed correspond to the iterations of the outer loop. This comparison has been realized when the source data set is Stanford1A segmented into 10 classes. The target data set is Stanford3A where we try to estimate the proportions of the 10 cell sub-populations.

Both algorithms have been implemented in Python. From our numerical experiments, we have found that the sequence of estimates $\hat{\pi}$ produced by the second algorithm described in 3.4.2 levels off approximately five times faster than the sequence of estimates produced with the descent-ascent procedure 3.4.1. This computational time gain can be accounted by the simple loop complexity of this second algorithm. However, in practice, the descent-ascent procedure seems to yield an estimate that is slightly more accurate than the estimate produced by the second procedure. Indeed, after enough iterations, the estimates of the descent-ascent procedure seem to get closer (according to the Kullback-Leibler divergence) to the benchmark π than the estimates of the min-max swapping procedure. Hence, for all the numerical experiments presented below, we report results produced by the descent-ascent procedure.

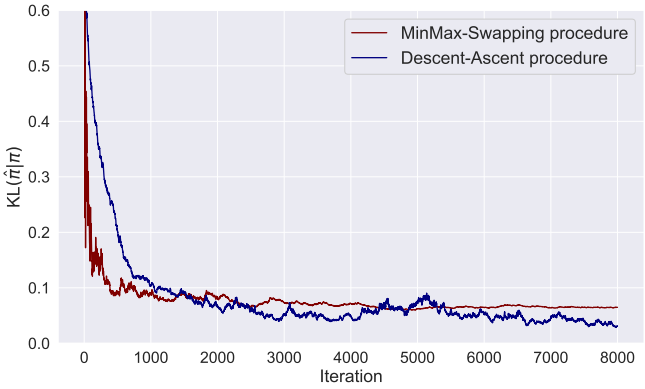


Figure 2: **Evolution of the Kullback-Leibler divergence between the estimated proportions $\hat{\pi}$ and the manual gating benchmark π.**

4 Application to real flow cytometry data analysis

4.1 Illustration of the methodology with a two classes example

In this section, we test and illustrate our proposed method in the setting where the cytometry data from HIPC described in Section 1.1 are divided only among two broad classes: the CD4 cells and the CD8 cells. For ease of visualization, we use only two markers: the CD4 marker and the CD8 marker – that is $d = 2$. This basic case, where two-dimensional data are divided into two classes, is a first illustration of our method in a favourable situation. We aim to demonstrate that our approach may reach a good approximation of the CD4 proportion and the CD8 proportion in a target cytometry data set with unknown gating.

We consider the two data sets from the HIPC T-cell panel that are displayed in Figure 3. The first data set is a series of cytometry measurements done in a Stanford laboratory on a biological sample that comes from a patient identified as patient 1. The Stanford data set is chosen as the segmented source data. The second data set is a series of cytometry measurements done in the same laboratory on a biological sample that comes from an other patient identified as patient 3. This second data set will be the target data set. The manual gating classification, and thus the class proportions are available for both samples. Nevertheless, we only use the classification available for the source (i.e. Stanford1A) data set in order to estimate the class proportions in the target (i.e. Stanford3A) data set. The manual gating classification for the target data set is only used as a benchmark to evaluate our method.

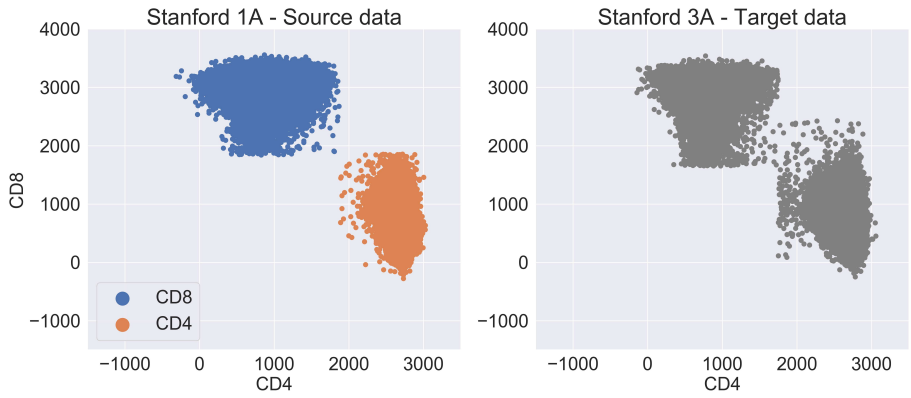


Figure 3: **Illustration of the `CytOpT` framework**. `CytOpt` estimate the class proportions in an unclassified data set (Stanford3A, right) from one classified data set (Stanford1A, left) without clustering the observations of Stanford3A.

First, to asses the relevance of our method, we present in Figure 4 the evolution of the regularized Wasserstein distance as a function of the weights associated to each class in the source distribution. To this end, we evaluate the function $F : h_1 \mapsto W^{\varepsilon}(\hat{\alpha}(h), \hat{\beta})$, where $h = (h_1, 1 - h_1)$, on a finite grid

$\mathcal{H} = \{h^{(1)}, ..., h^{(m)}\}$. For $h_1 \in \mathcal{H}$ we approximate $W^\varepsilon(\hat{\alpha}(h), \hat{\beta})$ by the estimator $\widehat{W}_n$ defined in equation (8). It can be observed that the regularized Wasserstein distance decreases as the class proportions of the source data set get closer to the class proportions of the target data set.

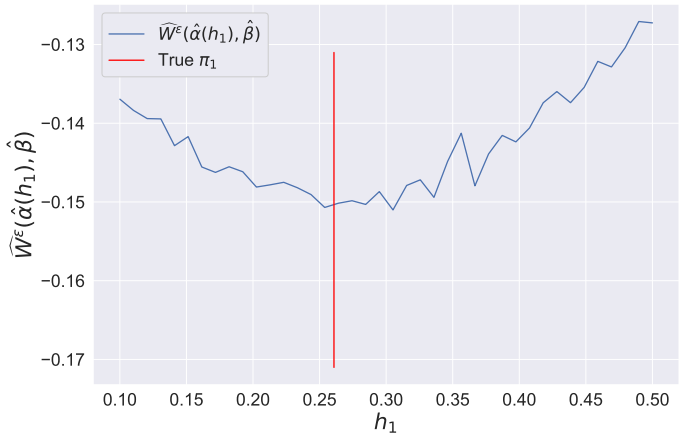


Figure 4: **Approximation of the function $h_1 \mapsto W^\varepsilon(\hat{\alpha}(h), \hat{\beta})$, for $h = (h_1, 1 - h_1)$, h_1 represents the weight associated to the CD8 cells.** The approximation of $W^\varepsilon(\hat{\alpha}(h), \hat{\beta})$ is produced using the estimator (8).

Using the segmentation of a cytometry data set, `CytOpT` adequately retrieves the true class proportions of an unlabelled cytometry data set. Even if the two classes situation is somewhat an easy scenario, one can observe the significant gap in the class proportions between the source data set and the target data set. In the Stanford1A data set the CD4 cells constitute 45.1 % of the cells and the CD8 cells 54.9 % of the cells, whereas in the Stanford3A data set the CD4 cells constitute 73.9 % (estimated at 73.3 % by `CytOpT`) of the cells and the CD8 cells 26.1 % (estimated at 26.7 % by `CytOpT`). Due to the stochastic nature of our algorithm, a new call to `CytOpT` could lead to a slightly different estimate $\hat{\pi}$ compare to a former estimation. However, the results displayed in Figure 7 and Figure 9 are representative of the quality of the estimation produced by `CytOpT` in such scenario.

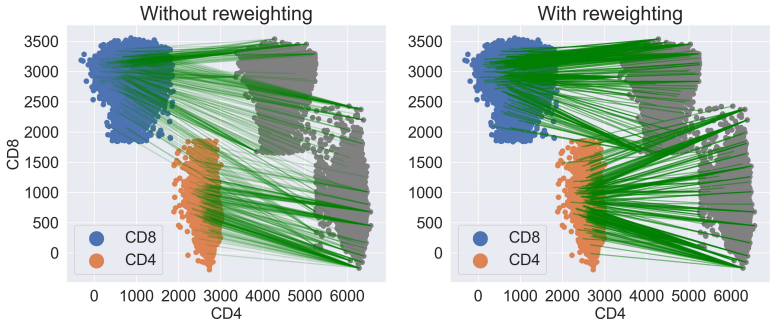


Figure 5: **Optimal transport plan P_ε between the source and target distribution.** A green line between X_i^s and X_j^t indicates that the optimal transport plan P_ε moves some mass from X_i^s to X_j^t. To facilitate readability, the target data set has been shifted, and only 500 coefficients of P_ε have been represented. Without re-weighting, mass from the CD8 class in the source data set is sent toward the CD4 class in the target data set. With re-weighting, the mass from one class in the source data set is sent to the corresponding class in the target data set.

The results presented in Figure 5 and Figure 6 represent the use of the soft-clustering and classification methods described in Section 3.2, and they outline the importance of re-weighting the source data. Without re-weighting, the classification obtained by optimal transport is really unlike the manual gating classification. However, one can re-weight the observations with the transformation $a(\pi) = \Gamma\pi$ in order to match the class proportions π in the target data set. Here, Γ is the linear operator defined in (15). Thus, when we re-weight our observations with the estimated class proportions $\hat{\pi}$ we get a much more interesting classification. Indeed, once the source data are re-weighted, the classification by regularized transport is very similar to the manual gating classification.

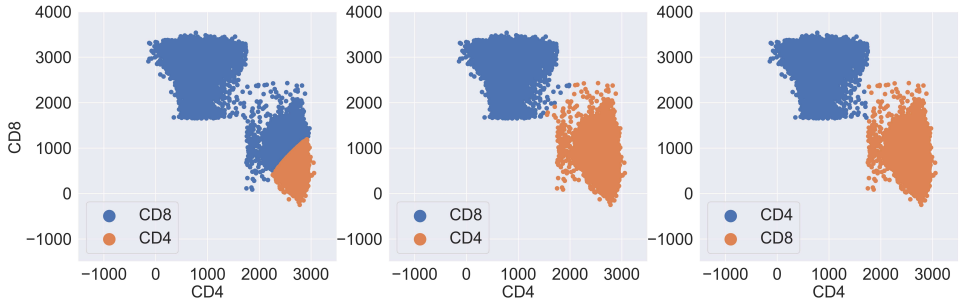


Figure 6: **Results of the soft assignment method and comparison with the manual clustering on the data set Stanford3A.** From left to right: Label transfer without reweighting - Label transfer with reweighting - Manual gating benchmark

4.2 Estimating 10 cell sub-type proportions from 7 cellular markers

We apply our method to the full T-cell panel of the HIPC flow cytometry data described in Section 1.1 in the Introduction. Contrary to the illustrative results presented previously Section 4.1, we now use all of the seven markers at hand to estimate the proportions of all 10 cell populations (a naturally harder task than for only 2 broad aggregated populations, as in Section 4.1). In a favourable case (target and source data set are from the same center), `CytOpT` results are very satisfactory as can be seen in Figure 7 where manual gating from patient Stanford1A was used as source to estimate the cell sub-types proportions from unlabelled cytometry measurements from an other patient (Stanford3C). Because the cytometry measurements came from two different patients, and according to the manual segmentation, the class proportions were widely different between the two data sets (compare Figures 1 and 7). In a more involved framework (i.e. when the target data set is from a different center), `CytOpT` still retrieves a good estimate $\hat{\pi}$ of the class proportions in spite of the large shift, see Figures 7 and 8. Finally, we provide a comprehensive evaluation of `CytOpT` performance in Figure 9, which features a Bland-Altman plot JM.Bland and DG.Altman [1986] displaying all the cell sub-types proportions estimations by `CytOpT` when targeting all available data sets across all the seven different included centers. In this example we solely used the reference classification from manual gating of one data set (Stanford1A) to estimate the class proportions in each of the targeted 61 unsegmented data sets. In more than 90% of the cases the difference between the estimated proportion and the manual gating benchmark proportion is below 5%. And in more than 99% of the cases the error is no more than 10%.

Finally, note that when applying our method to estimate class proportions on real flow cytometry data, a simple pre-processing is required. First, the signal processing of the cytometer can induce some contrived negative values of light intensity. To undo this effect, we merely threshold those few negative values at zero. Second, to settle the parameters of our algorithms, in particular ε, we need to bound the displacement cost. To do so, we scale the data such that: $\forall i \in \{1, ..., I\},\ X_i^s \in [0,1]^d$ and $\forall j \in \{1, ..., J\},\ X_j^t \in [0,1]^d$.

4.3 Comparison with `OptimalFlow`

`OptimalFlow` is a state-of-the-art method for performing supervised automatic gating recently developed by Barrio et al. [2019]. It is the first method to use the Wasserstein distance in the context of clustering flow cytometry data analysis. We stress that, even though both our method and `OptimalFlow` are supervised approaches that rely on the Wasserstein distance, they greatly differ. First, they have different objectives: `OptimalFlow` aims at a classifying the cells, while `CytOpT` directly aims at estimating the cellular sub-types proportions. Second, `OptimalFlow` uses the Wasserstein distance to cluster a compendium of pre-gated cytometry data sets and uses the notion of Wasserstein barycenter to produce a prototype data set for each cluster. Then, using the Wasserstein distance, `OptimalFlow` browses among the prototypes to extract the most relevant prototype to classify a test data set. But ultimately, the classification is performed with tools such as `tclust` B.Dost et al. [2010] or Quadratic Discriminant Analysis T.Hastie et al. [2009] that do not belong to the field of optimal transport. On the contrary, `CytOpT` only requires one pre-gated data set and the estimator of the class proportion is fully based on regularized optimal transport.

We have considered the comparison on two different cytometry panels: the HIPC panel presented

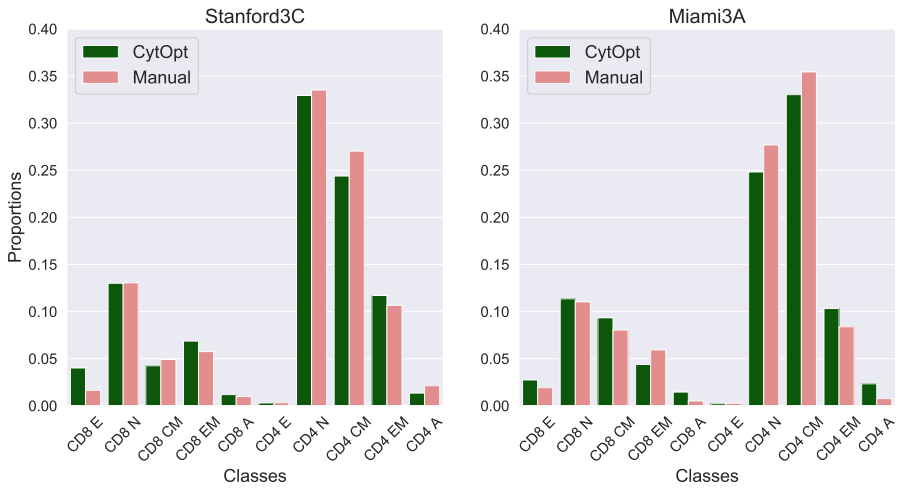


Figure 7: **Comparison of the estimated proportions $\hat{\pi}$ by `CytOpT` with the manual gating benchmark π.** Left: The target data set is Stanford3A. Right: The target data set is Miami3A. In both cases, the data set Stanford1A has been used as a source data set.

above, and the database that is used as an illustration in Barrio et al. [2019]. The second cytometry panel is available on GitHub[1], and we will refer to this panel as the OptimalFlow data sets.

The first comparison was performed on the HIPC data set. The comparison has been done in the following settings. The `OptimalFlow` learning database was composed of the nine Stanford cytometry data sets. `OptimalFlow` was set to cluster those nine data sets into 3 groups and to produce one prototype for each group. Thus, for each test data set, `OptimalFlow` browsed among those 3 prototypes to find the most relevant prototype to perform the classification. For the rest of the method, we applied `OptimalFlow` with the default settings proposed in the vignette[2]. As `CytOpT` requires a single learning data set, Stanford1A was used as the source data set. Figure 10 shows the results of this comparison on the replicate A of all the data sets available. For some data sets `OptimalFlow` slightly outperforms `CytOpT`. However, in a data set which present a significant shift compare to the learning data set `CytOpT` yields much better results than `OptimalFlow`. For instance, the data sets labelled 5, 11 and 17 in Figure 10, correspond to measurements performed in the Miami laboratory. As displayed on Figure 8, there is a spatial shifting between the Miami data and the Stanford data.

For the comparison on the OptimalFlow data, we have selected eight cell populations that were present in all the data sets. Then, we applied the two methods on the OptimalFlow data sets composed of those eight cell populations. In this second comparison, the `OptimalFlow` learning database was composed of the first three data sets of OptimalFlow data. Due to the smallness of the database, `OptimalFlow` was set to directly browse (without the clustering and prototype steps) among the three data sets of the learning database to extract the most appropriate learning data set. For `CytOpT`, we used the first data set as the source distribution. Figure 10 displays the results of this comparison. We obtain the same kind of results than those with the HIPC data set: if the test data set presents a spatial shifting that cannot be found in the learning database, `CytOpT` can handle this shift while it might be a difficult situation for `OptimalFlow`. One can acknowledge that for both comparisons `OptimalFlow` had access to a larger learning database than `CytOpT`.

5 Discussion

We have introduced `CytOpT`, a supervised method that directly estimates the cell class proportions, without a preliminary clustering or classification step. While obtaining a classification of the observation is often a mathematical tool for achieving automatic gating, we emphasize that from a clinical perspective it is the proportions of the different cell populations that matter. Indeed, this is the key parameter which informs on a patient's biological and health condition.

To further improve the performance of `CytOpT`, an additional pre-processing step to handle outliers could make `CytOpT` more robust to extreme cellular observations that have little biological meaning.

[1] at `https://github.com/HristoInouzhe/optimalFlowData`
[2] at `https://github.com/HristoInouzhe/optimalFlow`



Figure 8: **2D projection of two cytometry data sets.** Left: measurements performed in Stanford. Right: measurements performed in Miami. On this 2D projection one can asses the spatial shift induced by the technical variability of cytometry.

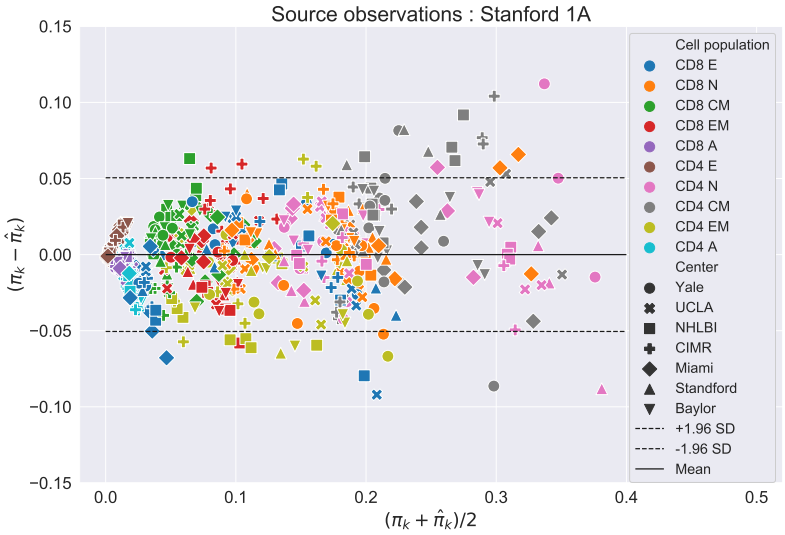


Figure 9: **Comparison of the proportions $\hat{\pi}$ estimated with `CytOpT` and the manual benchmark π on the HIPC database.** By plotting the difference against the mean of the results of two different methods, the Bland-Altman plot allows to assess the agreement between the two methods.

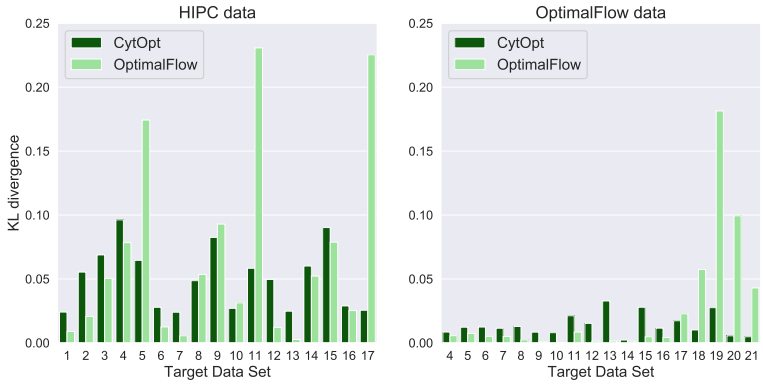


Figure 10: **Comparison between our algorithm `CytOpT` and one of the state of the art automated method for cytometry data analysis: `OptimalFlow`.** Left: Comparison on the HIPC data. Right: Comparison on the OptimalFlow data.

Indeed, due to the needed scaling standardization of the data, one outlier observation can change the layout of the source data in comparison with the layout of the target data. A data driven strategy to choose the regularization parameter ε could also help to tackle this outlier issue. Besides, the same method could be applied by taking into account the information of several segmented data sets, in order to derive the estimation of the class proportions in an unlabeled data set. In this case, a new estimator could be defined as $\hat{\pi} \in \arg\min_{h\in\Sigma_K} \sum_{m=1}^{M} W^{\varepsilon}(\hat{\alpha}^{(m)}(h), \hat{\beta})$, where $\hat{\alpha}^{(m)}(h)$ would be the empirical measure associated to the m^{th} source data set re-weighted such that the weight of the k^{th} class equals h_k for all $k \in \{1, ..., K\}$.

Extension of this work could consider the Sinkhorn divergence $S^{\varepsilon}(\alpha, \beta) = W^{\varepsilon}(\alpha, \beta) - \frac{1}{2}W^{\varepsilon}(\alpha, \alpha) - \frac{1}{2}W^{\varepsilon}(\beta, \beta)$, instead of $W^{\varepsilon}(\alpha, \beta)$. Indeed, the authors of J.Feydy et al. [2018] showed that S^{ε} has good theoretical properties that make it a suitable loss for applications in machine learning. Future research may also include a theoretical study of the convergence rates of the stochastic algorithms described in Section 3 that aim at solving optimization problem (11). Indeed, an improved optimization strategy could lead to practical advantages and alleviate the computational burden of `CytOpT`. Similarly, an appropriate adaptive decreasing step policy could lessen the variations of the results induced by the stochastic nature of our approach.

In conclusion, the proposed algorithm is very promising in terms of its performances in difficult situations where the technical variability of flow cytometry induces spatial shifts between samples. We think this new strategy is relevant for clinical applications, given the constant increase of the number of cellular markers used in cytometry analysis. This raises the question of the impact of such higher-dimensions, both for manual gating (dramatically increasing the resources necessary) and automated approaches (impacting the computational cost and efficiency). Regarding `CytOpT`, the increase in the number of markers does not extend the computational cost by much as the dimension only impacts the initial computation of the distance matrix between the observations. We argue that with higher-dimensional data, our method could perform even better as the Wasserstein distance would take advantage of all the dimensions available at once.

Codes and data availability

The illustrative HIPC data sets described in Subsection 1.1, as well as Python Notebooks to reproduce the Figures presented in this paper are available at `https://github.com/Paul-Freulon/CytOpt`.

References

N.Aghaeepour, G.Finak, H.Hoos, T.R.Mosmann, R.Brinkman, R.Gottardo, R.H.Scheuermann, FlowCAP Consortium, Dream Consortium, et al. Critical assessment of automated flow cytometry data analysis techniques. *Nature methods*, 10(3):228, 2013.

Y.Saeys, S.Van Gassen, and B.N.Lambrecht. Computational flow cytometry: helping to make sense of high-dimensional immunology data. *Nature Reviews Immunology*, 16(7):449, 2016.

D.Commenges, C.Alkhassim, R.Gottardo, B.P.Hejblum, and R.Thiébaut. cytometree: a binary tree algorithm for automatic gating in cytometry analysis. *Cytometry Part A*, 93(11):1132–1140, 2018.

Y.Ge and S.C.Sealfon. flowPeaks: a fast unsupervised clustering for flow cytometry data via K-means and density peak finding. *Bioinformatics*, 28(15):2052–2058, 05 2012. ISSN 1367-4803. doi: 10.1093/bioinformatics/bts300. URL `https://doi.org/10.1093/bioinformatics/bts300`.

B.P.Hejblum, C.Alkhassim, R.Gottardo, F.Caron, R.Thiébaut, et al. Sequential dirichlet process mixtures of multivariate skew t-distributions for model-based clustering of flow cytometry data. *The Annals of Applied Statistics*, 13(1):638–660, 2019.

H.Li, U.Shaham, K.P.Stanton, Y.Yao, R.R.Montgomery, and Y.Kluger. Gating mass cytometry data by deep learning. *Bioinformatics*, 33(21):3423–3430, 07 2017. ISSN 1367-4803. doi: 10.1093/bioinformatics/btx448. URL `https://doi.org/10.1093/bioinformatics/btx448`.

M.Lux, R.R.Brinkman, C.Chauve, A.Laing, A.Lorenc, L.Abeler-Dörner, and B.Hammer. flowLearn: fast and precise identification and quality checking of cell populations in flow cytometry. *Bioinformatics*, 34(13):2245–2253, 02 2018. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty082. URL `https://doi.org/10.1093/bioinformatics/bty082`.

E.del Barrio, H.Inouzhe, JM.Loubes, C.Matrán, and A.Mayo-Íscar. optimalflow: Optimal-transport approach to flow cytometry gating and population matching. *arXiv preprint arXiv:1907.08006*, 2019.

V.Brusic R.Gottardo, S.H.Kleinstein, M.M.Davis, D.A.Hafler, H.Quill, A.K.Palucka, G.A.Poland, B.Pulendran, E.L.Reinherz, et al. Computational resources for high-dimensional immune analysis from the human immunology project consortium. *Nature biotechnology*, 32(2):146, 2014.

H.Janati, M.Cuturi, and A.Gramfort. Wasserstein regularization for sparse multi-task regression. *arXiv preprint arXiv:1805.07833*, 2018.

R.Flamary, M.Cuturi, N.Courty, and A.Rakotomamonjy. Wasserstein Discriminant Analysis. *Machine Learning*, 107(12):1923–1945, December 2018. doi: 10.1007/s10994-018-5717-1. URL `https://hal.archives-ouvertes.fr/hal-01377528`.

M.Arjovsky, S.Chintala, and L.Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

J.Solomon, F.de Goes, G.Peyre, M.Cuturi, A.Butscher, A.Nguyen, T.Du, and L.Guibas. Convolutional wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Transactions on Graphics (TOG)*, 34(4):1–11, 2015.

M.Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems*, pages 2292–2300, 2013.

G.Peyré, M.Cuturi, et al. Computational optimal transport. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

A.Genevay, M.Cuturi, G.Peyré, and F.Bach. Stochastic optimization for large-scale optimal transport. In *Advances in neural information processing systems*, pages 3440–3448, 2016.

B.Bercu and J.Bigot. Asymptotic distribution and convergence rates of stochastic algorithms for entropic optimal transportation between probability measures. *Annals of Statistics*, To be published, 2020.

H.Robbins and S.Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

I.Redko, N.Courty, R.Flamary, and D.Tuia. Optimal transport for multi-source domain adaptation under target shift. *arXiv preprint arXiv:1803.04899*, 2018.

M.Ballu, Q.Berthet, and F.Bach. Stochastic optimization for regularized wasserstein estimators. *arXiv preprint arXiv:2002.08695*, 2020.

JM.Bland and DG.Altman. Statistical methods for assessing agreement between two methods of clinical measurement. *The lancet*, 327(8476):307–310, 1986.

B.Dost, C.Wu, A.Su, and V.Bafna. Tclust: A fast method for clustering genome-scale expression data. *IEEE/ACM transactions on computational biology and bioinformatics*, 8(3):808–818, 2010.

T.Hastie, R.Tibshirani, and J.Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

J.Feydy, T.Séjourné, FX.Vialard, SI.Amari, A.Trouvé, and G.Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. *arXiv preprint arXiv:1810.08278*, 2018.

J.M.Borwein and D.Zhuang. On fan's minimax theorem. *Mathematical programming*, 34(2):232–234, 1986.

C.Boyer, P.Weiss, and J.Bigot. An algorithm for variable density sampling with block-constrained acquisition. *SIAM Journal on Imaging Sciences*, 7(2):1080–1107, 2014.

A Two ways of solving the estimation problem

We will now present the technical details of the optimization strategies discussed in Section 3.4.

A.1 Descent Ascent procedure

In the optimization problem (11) that yields an estimator $\hat{p}$ of the class proportions, the weights of the source distribution are modified, but the support is fixed. Therefore, the parameterized measure $\alpha(h)$ is entirely represented through the mapping $a : h \mapsto a(h)$. Hence, we define the linear operator $\Gamma \in \mathbb{R}^{I\times K}$ that allows to express the transformation from the class proportions to the weight vector $a(h)$ as:

$$\forall (i,k) \in \{1,...,I\} \times \{1,...,K\},\ \Gamma_{i,k} = \begin{cases} \frac{1}{n_k} & \text{if } X_i^s \in C_k \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

Thanks to this transformation, the vector $a(h) = \Gamma h$ is such that the class proportions in the discrete measure $\sum_{i=1}^{I} a_i(h)\delta_{X_i^s}$ equals the vector h. To present the ideas of our first way to numerically solve (11), we re-write this optimization problem as:

$$\min_{h\in\Sigma_K} W^{\varepsilon}(\alpha(h),\beta) = \min_{h\in\Sigma_K} \max_{u\in\mathbb{R}^I} \mathbb{E}[g_{\varepsilon}(Y,u,h)], \tag{16}$$

where Y is a random variable with distribution β, and for $y_j \in \mathbb{R}^d$ an observation of the random variable Y,

$$g_{\varepsilon}(y_j,u,h) = \sum_{i=1}^{I} u_i a_i(h) + \varepsilon\left(\log(b_j) - \log\left(\sum_{i=1}^{I}\exp\left(\frac{u_i - c(x_i,y_j)}{\varepsilon}\right)\right)\right) - \varepsilon \tag{17}$$

We propose a gradient descent ascent algorithm in order to approximate $\hat{p}$ a solution of the optimization problem (11). At each iteration, our method runs multiple stochastic gradient ascent steps to estimate the solution of the inner maximization problem. Thanks to this estimation, we can approximate the gradient of the function $h \mapsto W(\alpha(h),\beta)$ that we wish to minimize.

We will now detail the optimization procedure. First, in order to avoid projecting h_n on the simplex Σ_K at each step of the procedure, we re-parameterized our problem with a soft-max function. Our new optimization problem is now:

$$\min_{z\in\mathbb{R}^K} W^{\varepsilon}(\Gamma(\sigma(z)),b) = \min_{z\in\mathbb{R}^K} F(z), \tag{18}$$

where $F(z) = W^{\varepsilon}(\Gamma(\sigma(z)),b)$, and $\sigma : \mathbb{R}^K \to \Sigma_K$ is the soft-max function defined as

$$\sigma(z)_l = \frac{\exp(z_l)}{\sum_{k=1}^{K}\exp(z_k)}. \tag{19}$$

Then, if we denote $\hat{z}$ a minimizer of (18), we will derive an estimator of p as $\hat{p} = \sigma(\hat{z})$.

Our approach uses the fact that the gradient of the function $a \mapsto W^{\varepsilon}(a,b)$ is given by:

$$\frac{\partial}{\partial a} W^{\varepsilon}(a,b) = u^* \tag{20}$$

where u^* is the unique solution to (3) centered such that $\sum_{i=1}^{I} u_i^* = 0$. This result is established in Proposition 4.6 of G.Peyré et al. [2019]. Hence, by applying the chain rule of differentiation, one obtains that:

$$\nabla_z W^{\varepsilon}(\Gamma(\sigma(z)),b) = (\Gamma J_{\sigma}(z))^T u_z^*, \tag{21}$$

where Γ is the linear operator defined in (15), u_z^* denotes the maximiser of (3) when the weights of the distribution α equal $a = \Gamma\sigma(z)$ and $J_{\sigma}(z)$ is the Jacobian matrix of σ, defined for $z \in \mathbb{R}^K$ by:

$$\forall (i,j) \in \{1,...,K\}^2,\ J_{\sigma}(z)_{i,j} = \begin{cases} \frac{\exp(x_j)\left(\sum_{k=1}^{K}\exp(x_k)\right) - \exp(2x_j)}{\left(\sum_{k=1}^{K}\exp(x_k)\right)^2} & \text{if } i = j \\ -\frac{\exp(x_i+x_j)}{\left(\sum_{k=1}^{K}\exp(x_k)\right)} & \text{otherwise.} \end{cases} \tag{22}$$

The stochastic algorithm that we consider to find a minimizer z^* of $F : z \mapsto W^{\varepsilon}(\Gamma\sigma(z),b)$ is then given by the recursive procedure

$$\hat{z}_{n+1} = \hat{z}_n - \eta\hat{\omega}(\hat{z}_n) \tag{23}$$

where $\hat{\omega}(\hat{z}_n)$ stands for a stochastic approximation of $\nabla_z W^\varepsilon(\Gamma(\sigma(\hat{z}_n)), b)$. We propose to estimate $\nabla_z W^\varepsilon(\Gamma(\sigma(\hat{z}_n)), b)$ by:

$$\hat{\omega}(\hat{z}_n) = (\Gamma J_\sigma(\hat{z}_n))^T \widehat{U}_{m+1}^{(n+1)} \tag{24}$$

that is we plug an estimate of $u^*_{\hat{z}_n}$ in the formula (21) of the gradient of the objective function when the weights of the source distribution equal $\Gamma(\sigma(\hat{z}_n))$

The estimate $\widehat{U}_{m+1}^{(n+1)}$ of $u^*_{\hat{z}_n}$ is calculated by the Robbins-Monro Algorithm H.Robbins and S.Monro [1951] which is defined through the recurrence:

$$\widehat{U}_{k+1}^{(n+1)} = \widehat{U}_k^{(n+1)} + \gamma_{k+1} \nabla_u g_\varepsilon(Y_{k+1}^{(n+1)}, \widehat{U}_k^{(n+1)}, \hat{z}_n) \tag{25}$$

where $\widehat{U}_0^{(n+1)}$ is arbitrary variable such that $\langle \widehat{U}_0^{(n+1)}, 1_I \rangle = 0$, and $Y_1^{(n+1)}, ..., Y_{m+1}^{(n+1)}$ are i.i.d random variables sampled from β at the iteration to pass from $\hat{z}_n$ to $\hat{z}_{n+1}$, and $(\gamma_n)_{n\geq 0}$ is a positive sequence of real numbers decreasing toward zero satisfying condition (7).

Once our minimization procedure is over and we have computed $\hat{z}$ a minimizer of (18), we compute an estimate of the class proportions in the target observations. To do so, we set

$$\hat{p} = \sigma(\hat{z}). \tag{26}$$

Choice of the parameters for the descent-ascent procedure For this Descent-Ascent procedure, we need to set several parameters. For the step size policy $(\gamma_k)_{k>0}$ of the inner loop, we rely on the recommendation done in B.Bercu and J.Bigot [2020]. Therefore, we chose $\gamma_k = \gamma/n^c$ where $c = 0.51$ and $\gamma = J\varepsilon/1.9$ with J the cardinal of β. We set the other parameters experimentally. Thus, we chose $n_{out} = 10000$, $n_{in} = 10$, $\eta = 10$ and $\varepsilon = 0.0005$.

A.2 Minmax swapping procedure

In this section we use the ideas of M.Ballu et al. [2020]. To propose an alternative scheme to solve the minimization problem (11) we slightly modify this problem by adding an entropic term

$$\varphi(h) = \sum_{k=1}^{K} h_k \log(h_k). \tag{27}$$

Thus, our new problem is now:

$$\min_{h\in\Sigma_k} W^\varepsilon(\alpha(h), \beta) + \lambda\varphi(h) = \min_{h\in\Sigma_k} \max_{u\in\mathbb{R}^I} \mathbb{E}[g_\varepsilon(Y, u, h)] + \lambda\varphi(h) \tag{28}$$

where Y is a random variable with distribution β, and g_ε is defined in (17).

By swapping the minimum and the maximum according to Fan's minimax theorem J.M.Borwein and D.Zhuang [1986] we get

$$\begin{aligned}
\min_{h\in\Sigma_k} \max_{u\in\mathbb{R}^I} \mathbb{E}[g_\varepsilon(Y, u, h)] + \lambda\varphi(h) &= \max_{u\in\mathbb{R}^I} \min_{h\in\Sigma_k} \mathbb{E}[g_\varepsilon(Y, u, h)] + \lambda\varphi(h) \\
&= \max_{u\in\mathbb{R}^I} \min_{h\in\Sigma_k} \sum_{i=1}^{I} u_i(\Gamma h)_i + A(u) - \varepsilon + \lambda\varphi(h) \\
&= \max_{u\in\mathbb{R}^I} A(u) + \min_{h\in\Sigma_k} \sum_{i=1}^{I} u_i(\Gamma h)_i + \lambda\varphi(h) - \varepsilon \\
&= \max_{u\in\mathbb{R}^I} A(u) + J_\lambda(u) - \varepsilon
\end{aligned} \tag{29}$$

where $A(u) = \varepsilon \sum_{j=1}^{J} \left(\log(b_j) - \log\left(\sum_{i=1}^{I} \exp\left(\frac{u_i - c(x_i, y_j)}{\varepsilon}\right)\right)\right) b_j$ and

$$J_\lambda(u) = \min_{h\in\Sigma_k} u_i(\Gamma h)_i + \lambda\varphi(h) \tag{30}$$

Arguing e.g. as in the proof of [C.Boyer et al., 2014, Proposition 4.1], the solution $h(u)$ of (30) is defined by

$$\forall k \in \{1, ..., K\},\ (h(u))_k = \frac{\exp\left(-\frac{(\Gamma^T u)_k}{\lambda}\right)}{\sum_{l=1}^{K} \exp\left(-\frac{(\Gamma^T u)_l}{\lambda}\right)} \tag{31}$$

By plugging (31) in (29), Problem (28) boils down to the following maximization problem with respect to $u \in \mathbb{R}^d$:

$$\max_{u\in\mathbb{R}^I} \varepsilon \sum_{j=1}^{J} \left(\log(b_j) - \log\left(\sum_{i=1}^{I} \exp\left(\frac{u_i - c(x_i, y_j)}{\varepsilon} \right) \right) \right) b_j - \lambda \log\left(\sum_{l=1}^{K} \exp\left(-\frac{(\Gamma^T u)_l}{\lambda} \right) \right) - \varepsilon \tag{32}$$

which can be rewritten

$$\max_{u\in\mathbb{R}^I} \mathbb{E}[f_{\varepsilon,\lambda}(Y, u)] \tag{33}$$

where Y is a random variable with distribution β, and for $y_j \in \mathbb{R}^d$ an observation of Y, and $u \in \mathbb{R}^I$,

$$f_{\varepsilon,\lambda}(y_j, u) = \varepsilon \left(\log(b_j) - \log\left(\sum_{i=1}^{I} \exp\left(\frac{u_i - c(x_i, y_j)}{\varepsilon} \right) \right) \right) - \lambda \log\left(\sum_{l=1}^{K} \exp\left(-\frac{(\Gamma^T u)_l}{\lambda} \right) \right) - \varepsilon \tag{34}$$

The fact that for all $y \in \mathbb{R}^d$ the function $f_{\varepsilon,\lambda}(y, .)$ is concave and the expectation form (33) of problem (32) lead us to estimate the optimal vector u^* by the Robbins-Monro algorithm given, for all $n \geq 0$ by

$$\widehat{U}_{n+1} = \widehat{U}_n + \gamma_{n+1} \nabla_u f_{\varepsilon,\lambda}(Y_{n+1}, \widehat{U}_n) \tag{35}$$

where the initial value $\hat{U}_0$ is a random vector which can be arbitrarily chosen, $Y_1, ..., Y_{n+1}$ are i.i.d random variables sampled from β ,and $(\gamma_n)_{n\geq 0}$ is a positive sequence of real numbers decreasing toward zero satisfying condition (7).

For $y_j \in \mathbb{R}^d$ an observation of Y, and $u \in \mathbb{R}^I$, the gradient of $\nabla_u f_{\varepsilon,\lambda}(y_j, u)$ is defined by $\forall i_0 \in \{1, ..., I\}$,

$$(\nabla_u f_{\varepsilon,\lambda}(y_j, u))_{i_0} = \frac{\sum_{l=1}^{K} \Gamma_{i_0,l} \exp\left(-\frac{(\Gamma^T u)_l}{\lambda} \right)}{\sum_{k=1}^{K} \exp\left(-\frac{(\Gamma^T u)_k}{\lambda} \right)} - \frac{\exp\left(\frac{u_{i_0} - c(x_{i_0}, y_j)}{\varepsilon} \right)}{\sum_{i=1}^{I} \exp\left(\frac{u_i - c(x_i, y_j)}{\varepsilon} \right)} \tag{36}$$

Once the algorithm has converged, and we get $\hat{U}$, a satisfactory approximation of a maximizer of problem 33, one can compute an estimate of the class proportions by setting

$$\hat{p} = h(\hat{U}), \tag{37}$$

where $h(u)$ is defined in 31.

Choice of the parameters for the min-max swapping procedure For this second procedure we have to set several parameters. To satisfy condition (7) for the step size policy $(\gamma_n)_{n>0}$ of the Robbins-Monro algorithm, we chose $\gamma_n = \gamma/n^c$ where $\gamma = 5$ and $c = 0.99$. The other parameters have been set empirically. We chose $\lambda = 0.0001$ and $\varepsilon = 0.0001$. The number of iterations for this stochastic ascent algorithm has been set to $n = 10000$.