Cytomulate: Accurate and Efficient Simulation of CyTOF data

Yuqiu Yang[1,2], Kaiwen Wang[1], Zeyu Lu[1], Tao Wang[2,3,*], Xinlei Wang[1,*]

1 Department of Statistical Science, Southern Methodist University, Dallas, TX, USA, 75275.
2 Quantitative Biomedical Research Center, Department of Population and Data Sciences, University of Texas Southwestern Medical Center, Dallas, TX, USA, 75390.
3 Center for the Genetics of Host Defense, University of Texas Southwestern Medical Center, Dallas, TX, USA, 75390.
* Corresponding authors

Short title: Cytomulate: Simulation of CyTOF data
Key words: CyTOF, simulation, proteomics

Abstract:

Cytometry by time-of-flight (CyTOF) is a proteomic analysis technique capable of providing information on millions of cells. Along with technological development, researchers have devised many data analysis tools to offer deeper insights into the CyTOF data at the single-cell level. However, objective evaluations of these methods remain absent as most evaluations are conducted via real data where the ground truth is in general unknown. Critically, there lacks a dedicated simulation method for CyTOF to generate simulated datasets, with true labels against which model performance can be measured. In this paper, we develop Cytomulate: an accurate, efficient, and well-documented simulation model and algorithm for reproducible and accurate simulation of CyTOF data, which could serve as a foundation for future development and evaluation of new methods. With its two simulation modes: creation mode and emulation mode, Cytomulate is capable of simulating datasets that capture various characteristics of CyTOF data under different circumstances. We also provide the users with means to simulate batch effects, temporal effects, and cellular trajectories. Empirical experiments suggest that Cytomulate is superior in learning the overall data distribution than single-cell RNA-seq-oriented methods such as Splatter and generative models like LAMBDA.

Background:

Recent years have seen exponential growth in applications of mass cytometry (CyTOF) which characterizes proteomics profiles of tens of thousands or millions of heterogeneous single cells on 30-40 protein channels with minimal spill-over effects (1–3). As the technique of CyTOF matures, many data analysis methods have been proposed to offer insights into various aspects of single-cell experiments (4). For example, in (5), the authors have observed that the signal intensity of a CyTOF experiment would vary over time due to changes in instrument performance. Their proposed method is currently the dominant tool to alleviate the temporal effect using polystyrene beads embedded with metal lanthanides. On the other hand, methods such as CytofRUV (6), and Cytonorm (7), etc. aim at getting rid of batch effects to facilitate comparisons across different CyTOF experiments. Deep learning algorithms such as residual networks (ResNet) (8) and generative adversarial networks (GAN) (9) were also adopted to resolve the same problem. While some software such as FlowSOM (10), LAMBDA (11), and Bayesian trees (12) focus on automatic cell type identification, others like CytoTree (13) attempt to explore the trajectories along which cells differentiate into other cells. Various data visualization and dimension reduction algorithms have also been applied to CyTOF (14). Typical examples include: t-SNE (15), UMAP (16), MDS (17), SAUCIE (18), scvis (19), etc. As the output of a CyTOF acquisition software is in Flow Cytometry Standard (FCS) format, packages such as flowCore (20) and Seurat (21) have incorporated standardized and efficient pipelines to read, write, and preprocess CyTOF data.

Despite the vast array of methods applicable to CyTOF, most of the CyTOF analysis methods developed by researchers are often evaluated against real data. Without a gold standard and known truth, such as labels and actual distributions, performance assessment becomes somewhat subjective while also blurring the line between method validation and result discovery. Simulated datasets, albeit being synthetic, contain ground truth against which various methods can be tested. Therefore, a simulation tool naturally provides a framework for a systematic evaluation of different methods. By varying parameters or various settings of key factors, researchers can easily generate an unbiased coverage of the system under study in a straightforward manner. On the other hand, real experiments require detailed prior planning, precise executions, and potentially abundant resources. This makes simulation a vital tool in methodological development. However, there is a lack of formal simulation tools for CyTOF data, which serve as the foundation for developing and validating all other CyTOF data analysis tools. By contrast, various methods have been developed for simulating single cell RNA-sequencing (scRNA-seq) data, such as Simple (22), Lun (23), BASICS (24), and Splatter (25), among which Splatter is dominant.

Since both scRNA-seq and CyTOF are single-cell sequencing techniques, it might be tempting to tweak scRNA-seq simulation tools such as Splatter and apply them to CyTOF. However, such a

process can be highly non-trivial and the end results may be unsatisfactory if only small adjustments are sought after. First of all, these two techniques are of different biological nature. Single-cell RNA sequencing (scRNA-seq) examines the expression of thousands of RNAs of several thousand cells, whereas CyTOF captures information on typically 30-40 proteins for tens of thousands or even millions of cells. Therefore, compared with scRNA-seq, CyTOF is more closely related to clinical phenotypes and is more likely to identify rare cell populations. Furthermore, while technique drop-out is a common issue in scRNA-seq data, it is a much less encountered problem in CyTOF experiments. What's more, the expressions in scRNA-seq are integer-valued "counts", in contrast to real-valued data or even negative values in CyTOF expression matrices due to noise and normalization by the inverse hyperbolic sine function with a cofactor of 5 (26), which, for the sake of clarity, will be referred to as asinh-transformation herein and after. Consequently, a sequence of nonlinear transformations is almost inevitable to render the expression matrices acceptable inputs to simulation tools like Splatter, which will yield biased results. Finally, the expression levels of various protein channels in CyTOF are naturally dependent. Extending the discrete random variable model such as the Poisson-Gamma mixture model adopted by Splatter to capture the dependency structures present in CyTOF might not be a trivial task. To bridge this gap, we develop Cytomulate, the first formal simulation tool that is capable of generating simulated datasets that capture various characteristics of CyTOF data. Further, our framework combines the flexibility of generating complex, realistic data with efficiency and usability, allowing researchers to acquire data at scale and practitioners to easily simulate datasets for their workflows.

Results:

Characteristics of CyTOF data:

To motivate the design choices of Cytomulate, we have explored prominent characteristics of CyTOF using various public datasets and in-house datasets from UTSWMC (e.g., Finck (5), Levine_13dim (27), Levine_32dim (27), CytoNorm (7), CyAnno (28), Covid (14)). In this section, we mainly used the protein expressions associated with the first patient in the dataset Levine_13dim which is publicly available via the R package HDCytoData (29) for the purpose of demonstration. As the protein expressions have already been pre-processed to get rid of artificial effects, only asinh-transformation was applied.

High volume, high dimension, and heterogeneity: The expression matrix associated with a typical CyTOF experiment usually contains tens of thousands or even millions of cell events (rows), 30-40 protein channels (columns), along with 10-30 different cell types. For example, the data of the first patient in Levine_32dim has 72,463 cell events, 32 columns, and 14 cell types manually gated by the authors. Therefore, efficiency in both extracting major patterns from

numerous cell events and generating a large dataset should be crucial for any potential simulation tools.

Zero-inflation: Like scRNA-seq data, the protein expressions observed in a CyTOF experiment often contain featuresthat are abundant with zeros or values close to zero. Biologically, these columns correspond to surface proteins that are often too scarce to be detected by the machine. Take, for example the CD4 T cells in the dataset, we see in Fig. 1a and Fig. 1b that the majority of the columns contain a high proportion of values close to zero.

Dependent columns: In Fig. 1c, we calculated the correlation matrix among all the protein channels using the data of CD4 T cells. Clearly, markers such as CD44 and CD45 are correlated, implying that the independence assumption between columns might be too restrictive. This suggests modeling the data jointly instead of marginally.

Normality: To explore the data normality, we performed dimension reduction using Principal Component Analysis (PCA). If the data is Gaussian or near Gaussian, then the PCs would look somewhat elliptic. In Fig. 1d, we see that all cell populations are roughly elliptic, suggesting the usage of Gaussian distributions as basic building blocks.

Temporal effect: As illustrated in (5), the signals of a single CyTOF experiment would typically vary over time due to changes in instrument performance. Since Cytomulate aims at modeling asinh-transformed protein expressions, unlike in (5), we chose to demonstrate the signal drift on transformed data (Supp. Fig. 1a). As a first attempt to model the signal drift, we centered each signal, fitted a smoothing spline against one of them, and subtracted the fitted value from each signal. In Supp. Fig. 1b, we see that the adjusted signals overlap and are roughly homoscedastic, suggesting that the overall shape of the signal drift (temporal effect) may be shared by all cell types within a single batch and that we could model the effect using an additive model.

Batch effect: Apart from the temporal effect, many researchers also reported the presence of batch effects across multiple CyTOF experiments due to sample collection, stimulation, staining, etc (30). In (7) they reported that not only are the batch effects cell-type specific, they also vary depending on protein channels. In Supp. Fig. 1c, we illustrated this phenomenon by an interaction plot using the protein channels of CD4 T cells and CD8 T cells measured on two patients in the dataset Levine_32dim (29). We see that not only do expression levels differ between two batches, the differences also vary by cell types and protein channels. This implies that in addition to the main batch effect, we should also take into consideration the interaction effects.

Cellular trajectory: Similar to scRNA-seq, a common task is to study cellular trajectory where the expressions of an individual cell are assumed to vary according to some continuous path that

connects one cell type to another. In the context of scRNA-seq, lots of methods have been developed to align cells on continuous paths with typical examples Slingshot and Monocle (31–33). To our knowledge, CytoTree (13) is the only available method of trajectory inference designed for CyTOF. For the purpose of demonstration, we regenerated the differentiation paths following their online tutorial in Supp. Fig. 1d.

Cytomulate Framework

In light of the previous exploratory data analysis, we propose Cytomulate, an accurate and efficient simulation algorithm of CyTOF data based on Gaussian Mixture Models (GMM) (34). It can easily extract prominent characteristics from a large number of cell events commonly encountered in CyTOF experiments; it is capable of generating multiple realistic looking synthetic datasets within seconds; and it has a detailed documentation elucidating how all the components work together.

Cytomulate provides users with two simulation modes: creation mode and emulation mode.

Creation mode aims at quickly generating synthetic datasets that capture some aforementioned characteristics of CyTOF data such as zero-inflation, column dependencies, and near normality. As this mode does not intend to extract the data distribution of some specific dataset as a reference, it is particularly useful for testing methods on a large scale. For example, in (14), Wang et al. used this mode of Cytomulate to evaluate the performance of 20 dimension reduction methods from various aspects such as local and global structure preservation on 425 simulated datasets with different attributes (e.g., different numbers of cell types and numbers of differentiation paths). This mode allows practitioners to precisely define each simulated dataset with known ground truth, which is not often possible with real data. Further, it can also serve as a pedagogical tool for researchers who are not familiar with the format of a CyTOF dataset.

Emulation mode, on the other hand, is designed to learn the distribution of some real CyTOF data. Hence, it requires as inputs the expression matrix of one particular CyTOF dataset to mimic and the associated cell type labels. Although Cytomulate can potentially perform cell identification or clustering using common algorithms of the likes of K-means, we leave the choice of such algorithm for cell typing to our users as a design choice for the following two reasons: 1. According to (35) and (36), the k-means algorithm and GMM-based model (SWIFT) are often not the optimal choice for clustering CyTOF data; 2. There are already plenty of well-established cell identification algorithms and automatic gating techniques such as FlowSOM (10), Bayesian Tree (12), ACDC (37), SCINA (38), etc, all of which have similar goals but slightly different design considerations. For example, using FlowSOM requires minimal prior knowledge on the cell population other than some crude idea on the number of cell types. On the other hand, Bayesian Tree and ACDC use a template as an input to facilitate the

automatic gating process. Since we expect the users of the emulation mode to be somewhat experienced in the field, we think it would be better to let the users choose and apply their own cell identification algorithms that suit their situations the best. The resulting expression matrices of this mode are thus expected to be a closer approximation to the real data, as opposed to those generated by the creation mode.

To dive into the details, Cytomulate, at its core, uses a GMM with truncation below at 0 to approximate or generate the probability density of a given cell type. Although the exploratory data analysis in the previous section shows that a multivariate Gaussian might be a good candidate for modeling a specific cell type in a CyTOF experiment, this should be taken with a grain of salt since the clustering in Levine_32dim is quite detailed. However, we should not expect all users to be experts in identifying granular cell types. For example, there are 4 subtypes of B cells in Levine_32dim: mature B cells, plasma B cells, pre-B cells, and pro-B cells. If the users are only interested in B cells a single multivariate Gaussian would be inadequate in describing the data generating mechanism (see the comparison study in the next section for details). Therefore, we chose GMM for its flexibility and its mathematical tractability.

To cater for complex needs of methodological developments for CyTOF data, in both simulation modes, users are allowed to vary cell abundances, noise levels, batch effects, temporal effects, as well as cellular trajectories. The simulated batch effects, temporal effects, cellular trajectories, as well as the noise are then added once we sample from the GMMs. In Fig. 2, we outlined the framework of Cytomulate. Further details on the two simulation modes and the probabilistic model can be found in the Methods section.

We also modularized all the aforementioned functionalities so that each of them can be easily fine-tuned or turned off, according to an user's specific need. Cytomulate returns the final expression matrices, the cell type labels associated with the expression matrices, as well as cellular trajectory information for each cell event. In the following sections, we conduct some comparison studies and demonstrate the usage of Cytomulate through examples.

Simulation Comparison Study

To the best of our knowledge, Cytomulate is the first formal simulation tool designed specifically for CyTOF experiments. Nevertheless, we still wish to demonstrate through a series of empirical experiments and examples the flexibility, the strengths, as well as some potential future improvements of Cytomulate.

The emulation mode is used throughout the comparison study, where the following simulation models are included:

1. Splatter (25)

2. LAMBDA (11)
3. Cytomulate constrained to one Gaussian component with diagonal covariance per cell type (IG)
4. Cytomulate constrained to one Gaussian component with full covariance per cell type (DG)
5. Cytomulate with full covariance and model selection via Bayesian Information Criterion (BIC) from GMMs with components varying from 1 to 9 (Cytomulate)

Below we give a brief description of Splatter and LAMBDA followed by the rationale behind the choice of the models to be compared.

Splatter, or specifically Splat, is originally designed for simulating single-cell RNA sequencing data. The model uses a Gamma-Poisson mixture as a base to simulate counts observed in scRNA-seq experiments. The authors also take into consideration outliers, library size, mean-variance trend, and technical dropout. As Splatter expects integers as inputs, to simulate CyTOF data which usually have been transformed into a numeric scale using the inverse hyperbolic sine function with a cofactor of 5, we can carry out its inverse transformation, round up the results to the nearest integer (26), apply Splatter to simulate counts, and transform the simulated expressions back into the numeric scale.

LAMBDA (Latent Allocation Model with Bayesian Data Analysis) is a Bayesian hierarchical model designed for automatic cell identification in CyTOF data. The model assumes that given a cell type, the cell expressions follow a zero-inflated multivariate Gaussian distribution. The cell types, on the other hand, are generated from a Categorical distribution whose parameters are controlled by some clinical information. Parameter estimation is accomplished by a stochastic EM algorithm. It is worth pointing out that LAMBDA is not a simulation tool and its simulation functionality is only a byproduct of LAMBDA. We included the method here because of its rigorous parameter estimation procedure, which would provide us with a yardstick against which we can evaluate the quality of the simulated datasets in terms of the first and second moments.

We hope that the comparison between Splatter and Cytomulate would convince the readers that adapting a simulation tool designed for scRNA-seq data to CyTOF data is nontrivial. Otherwise, the end results may not be satisfactory. By taking into consideration the linear dependencies among protein channels, the DG model should show the value of joint modeling when compared with the IG model. Compared with the DG model, our core model, Cytomulate, should demonstrate the merit of modeling non-normality.

To demonstrate the versatility of Cytomulate, six publicly available datasets from different species and diverse anatomic sites are collected: Levine_32dim (27), Levine_13dim (27), Samusik (39), CyAnno (28), Covid (14), and LG (40). If multiple samples exist for one dataset,

the first sample is used. See Table 1 for details. All of these datasets have been bead-normalized and transformed by inverse hyperbolic sine function with a cofactor of 5. Cell type information is also publicly available for the first four datasets. Cell types in Covid and LG were identified via FlowSOM and manual gating (14).

Cytomulate well approximates real data

In this comparison study, we make use of all six datasets and include as candidate models IG, DG, Cytomulate, LAMBDA, and Splatter. Three metrics are adopted to quantify the quality of the simulated datasets: 1. L2 distance between the simulated mean and the observed mean; 2. Frobenius distance between the simulated covariance and observed covariance; 3. Kullback-Leibler (KL) divergence between the simulated data and observed data. See the Methods section for further details. For each dataset, the process is repeated 50 times for 50 simulated datasets.
We first show a detailed comparison using the dataset CyAnno. In Fig. 3, we see that Cytomulate outperforms all other methods benchmarked in this study with the smallest KL divergence. As the KL divergence is a measure of how similar two distributions are, we conclude that Cytomulate is able to yield a better approximation to the real data compared with other methods. For mean estimation, all methods except Splatter work well, with LAMBDA achieving the smallest MSE as expected, closely followed by Cytomulate. We are surprised to see that Cytomulate outperformed LAMBDA in estimating the covariance structure. This result also empirically demonstrated that improvements can be made if only the first two moments are considered. Splatter, due to the usage of nonlinear transformations and marginal modeling, performs the worst in all benchmarks by an overwhelmingly wide margin, demonstrating that Splatter is suboptimal when simulating CyTOF data.

We further applied UMAP to illustrate the resulting expression matrices of each method (Fig. 4a - Fig. 4d, Supp. Fig. 2). Due to the granularity of the cell type labels included in the dataset, at the first glance, results returned by Cytomulate and LAMBDA resemble the real data. However, if we examine the blobs at the bottom right of the central cluster more closely, we see that Cytomulate depicts the data in far more detail than does LAMBDA. The preservation of small clusters provides an advantage in preserving rare cell types and their unique phenotypes, whereas LAMBDA's resolution limitations tend to group these cells into adjoining clusters.

We now provide the rankings on the three metrics for all five datasets (Fig. 4e). By using GMMs at the cell type level, Cytomulate always gives the best approximation to the real data as measured by the KL divergence. As expected, by formally deriving a rigorous estimation procedure, LAMBDA is able to achieve the highest ranking in terms of mean and sometimes covariance. However, as we have seen previously, the sufficiency of only considering the first two moments highly depends on the data and the coarseness of the cell type labels. It is also

evident that Splatter always gives the worst performance due to the bias introduced by nonlinear transformations. Even working with Independent Gaussian would yield a performance boost.

Cytomulate is efficient

As a single CyTOF experiment could contain tens of thousands or millions of cell events along with 30 - 40 protein channels, computational efficiency including estimation and simulation is a significant issue. We conducted an analysis on the running time of each method by varying the number of cells and the number of markers using the CD8 Naive cells in the dataset CyAnno which contains 8,414 cell events and 39 protein channels. To be more specific, while evaluating the impact of sample size on estimation time and simulation time, we used all 39 protein channels and applied bootstrap sampling to generate expression matrices containing various numbers of cell events. We fixed the number of cell events and sampled the columns to be included when investigating the effect of the number of markers on processing time. We included IG, DG, Cytomulate with 5-component GMMs (Restricted Cytomulate), Cytomulate, LAMBDA, and Splatter. The experiment was carried out on a MacBook Pro with a 2.9 GHz Quad-Core Intel Core i7 processor. Since Cytomulate is implemented in Python while Splatter and LAMBDA are R packages, we used Python's time module to measure the performance of Cytomulate-related methods and Sys.time function in R to time Splatter and LAMBDA.

Overall both estimation time and simulation time for all methods under consideration appear to be polynomial with respect to the number of cells and the number of protein channels. In Supp. Fig. 3a, we see that for a moderate-size dataset containing 100,000, even Cytomulate with model selection from 9 models only takes around 2 minutes. If restrictions are imposed, the estimation can be done within seconds. On the other hand, LAMBDA, due to the usage of a stochastic EM algorithm for the model estimation, is twice as slow. It's also surprising to see that the number of markers has a large impact on the estimation time of LAMBDA compared with other methods (Supp. Fig. 3b). In terms of simulation time (Supp. Fig. 3c and Supp. Fig. 3d), we see that Cytomulate is always the most efficient under both circumstances with or without model selection, while Splatter seems the least efficient.

Overall, Cytomulate is capable of handling large datasets commonly encountered in CyTOF experiments.

Cytomulate is robust against cell-type resolution

Although an array of cell type labels is a required input for the emulation mode, we do not expect high granularity of the labels. In fact, different datasets and analyses call for different cell

typing approaches: a large dataset with only B cells may warrant more detailed cell types to reveal their differentiation stages, whereas small whole blood samples may not afford the same resolution with reasonable cluster sizes. In this experiment, we demonstrate that even if the cell identification is not of high resolution, Cytomulate is still able to yield a satisfactory approximation of the data distribution. We included Independent Gaussian, Gaussian, Cytomulate, LAMBDA, and Splatter.

As we mentioned previously, there are 4 subtypes of B cells in Levine_32dim dataset: mature B cells, plasma B cells, pre-B cells, and pro-B cells. In the first setting, we provided every method with these detailed cell type labels. In the second setting, however, we amalgamated all the labels into “B cells“ and fed this information to the methods.

We used UMAP to visually compare the results under two settings (Fig. 5). Although with the detailed labels, all methods except for Splatter seem to give acceptable approximations to the real data (Fig. 5 and Supp. Fig. 4), only Cytomulate is able to model the 4 major clusters and 1 small cluster present in the real data in the absence of detailed cell type information (Fig. 5b). For instance, if detailed information is not given, LAMBDA approximates the entire dataset with one Gaussian distribution which puts its mode around the middle of the two separated major clusters where the actual density is low (Fig. 5d).

To quantify the comparisons, we repeated the process 50 times and computed the KL divergence (41) of the simulated datasets of each method under two settings (Fig. 5f). Although more detailed cell types would consistently improve the overall performance, Cytomulate appears to be the most robust among all methods. In fact, both settings of Cytomulate outperform all other methods regardless of cell-typing details.

Contemplating on the results of this series of comparison studies, we see that Cytomulate is well suited for the large number of cell events commonly encountered in CyTOF experiments, is robust to upstream cell type identification results, and accurately captures the prominent characteristics of the data.

Complex simulations:

In this section, we briefly demonstrate how the basic Cytomulate model used in the comparison study can be extended to incorporate more complex effects commonly encountered in CyTOF experiments such as batch effects, temporal effects, as well as cell differentiations.

Temporal effects:

As was noticed in (5), due to cell debris, etc. the sensitivity of a CyTOF machine may vary over time during data generation, causing signals for cells from one batch to fluctuate. In Cytomulate, we assume that the temporal effect for each batch is generated by a function defined on the unit interval such that the value at time zero is always zero , representing that there is no temporal effect at the beginning. We provide users with 3 options to simulate a temporal effect, each with a different level of control over the overall shape of the temporal effect:

1. Brownian Bridge (42) : The shape of the entire trajectory is randomly generated from a Brownian Bridge where the ending value is randomly sampled from a normal distribution. To evaluate the function on the unit interval, the Akima spline (43) is used for interpolation. This option gives the users minimal control over the evolution of the temporal effects.
2. Polynomial: This option allows the user to specify the rough shape of the trajectory via the coefficients of a polynomial. A linear transformation will be used to make sure the resulting temporal effect starts at 0 and ends at a value randomly generated from a normal distribution.
3. Spline: During the preprocessing steps, it is possible that the users can extract the temporal effect component. The spline option permits the users to add the original temporal effect back to the simulated dataset via a smoothing spline. Since the size of the simulated dataset might differ from that of the original dataset, Cytomulate will automatically rescale the time interval so that the entire trajectory could be retained in the simulated dataset (Fig. 6a).

Batch effects:

As was noticed in (7), the batch effects in CyTOF experiments vary not only by batches but also by cell types and protein channels. To simulate batch effects, we decompose the effects into overall batch effects which assume different values for different batches and local batch effects which vary depending on the cell types and protein channels in a similar fashion to the way we decompose effects into main effects and interactions in Analysis of Variance (ANOVA). Further details can be found in the Methods section. As a demonstration, we simulated two batches with 3 cell types using the creation mode of Cytomulate (Fig. 6b).

Cellular trajectory differentiation:

To simulate cell differentiations, Cytomulate adopts a two-step method. In the first step, a polytree (a directed acyclic graph whose underlying undirected graph is a tree) or a polyforest (a collection of nonoverlapping polytrees) is generated (44). In the Creation Mode, this is done by first randomly grouping the cell types and applying the Kruskal's algorithm (45) with random weights to sketch the underlying tree or forest. In the Emulation Mode, this is accomplished by the Clauset-Newman-Moore greedy modularity maximization algorithm (46) (47) followed by

the Kruskal's algorithm with the L2 distances among observed average cell means as weights. In both circumstances, to convert the graph to a directed acyclic graph (DAG), a depth-first search with a randomly selected node as the root on each graph component is performed. Inspired by Splatter (25), in the second step, we use a series of Brownian Bridges to generate the actual trajectories along the polytree or the polyforest. For illustration, we used the creation mode of Cytomulate to generate 5 cell types aligned on one polytree (Fig. 6c).

Discussion and conclusions

Simulation is a powerful, flexible and cost-effective tool for performance evaluation, which plays a vital role in methodological development. A common way to test analysis methods is to conduct simulation studies, where generating multiple datasets, as many as we wish, with different parameters or assumptions, can be quickly done at least cost. However, to the best of our knowledge, there is no formal method available for accurately synthesizing CyTOF data.

In this paper, we introduced Cytomulate, a data analysis tool designed for simulating CyTOF data accurately and efficiently. It builds on Gaussian Mixture Models (GMM) with a fast and robust estimation procedure and appealing mathematical properties. Further, Cytomulate features a well-defined probabilistic model to capture prominent aspects of CyTOF data while also incorporating two simulation modes: creation mode and emulation mode to cater to the needs of various researchers. By examining various aspects of Cytomulate such as estimation efficiency, simulation efficiency, tolerance towards coarse upstream cell type identification, and resemblance to real data using six publicly available datasets, we found that Cytomulate is superior in approximating the overall data distributions regardless of the granularity of the input. We also illustrated the versatility of Cytomulate by simulating more complicated phenomena commonly encountered in a CyTOF experiment such as signal drifts (temporal effects), batch effects, and cell differentiations. Our software package is not only well documented with detailed tutorials and guidelines hosted online (cytomulate.readthedocs.io) but also well maintained with clearly defined software support cycles, code assurance workflows, and an open source license. Cytomulate is currently available on both PyPI and conda for installation via standard procedures.

One limitation of this study involves the estimation procedure we chose for the emulation mode. As a trade-off for efficiency, various aspects of a CyTOF data such as means, covariances, and cell differentiations are captured in a sequential manner instead of being modeled jointly. Consequently, although the GMMs provide a reasonable approximation to the data distribution of CyTOF due to its universal approximation to densities (48), the estimated parameters are necessarily biased. Due to the sheer number of cells typically encountered in CyTOF and the complexity of the model, traditional inference methods such as Markov Chain Monte Carlo (MCMC) might not ameliorate this situation. Fortunately, tremendous progress has been made in

the field of deep generative models, which has been successfully applied to tasks such as estimating densities and synthesizing realistic-looking images on large datasets (49,50). Some typical examples include normalizing flow (51), variational auto-encoder (52), and generative adversarial networks (53). Exploiting said methods to gain deeper insights into CyTOF datasets thus could be a possible future direction.

Methods

Probabilistic model of Cytomulate:

The probabilistic model of Cytomulate is intended to model the $arcsinh(./5)$-transformed CyTOF data. Suppose that we have $b = 1, \cdots, B$ batches, $m = 1, \cdots, M$ protein channels, $n = 1, \cdots, N^b$ cell events for the bth batch, and $p = 1, \cdots, P$ cell types. We denote the probability of observing the pth cell type in the bth batch by π_p^b. Then, for the nth cell event in the bth batch, we assume that the cell type is sampled from $\Pi_n^b \sim Categorical(\pi^b)$, where $\pi = [\pi_1^b, \cdots, \pi_P^b]$ is a P-by-1 probability vector.

Suppose the cell type for the nth cell event in the bth batch is p. Then, we assume that, without cell differentiation or background noise, the latent expression is $X_n^b \sim f^p(\mu^p, \Sigma^p)$, where $f^p(\mu^p, \Sigma^p)$ denote some distribution with mean μ^p, and variance-covariance matrix Σ^p. We use a Gaussian Mixture Model to approximate this distribution in Cytomulate.

The simulation of cell differentiation consists of two steps. Suppose cell type p differentiates into z^p "children" $p^{c_1}, \cdots, p^{c_{z^p}}$. Then, similar to the cell type generation, we assume that for the nth cell event whose cell type is p, the cell type it differentiates into is sampled from $\Omega_n^p \sim Categorical(\omega^p)$, where $\omega^p = [\omega_1^p, \cdots, \omega_{z^p}^p]$ is a z^p-by-1 probability vector.

Given the realization for the nth cell event is p^c, we construct the differentiation trajectory $G^{p \to p^c}: [0, 1]^M \mapsto R^M$ whose components are sampled from a series of Brownian bridges such that the mth marker satisfies $G_m^{p \to p^c}(0) = 0$, and $G_m^{p \to p^c}(1) = \mu_m^{p^c} - \mu_m^p$, where μ_m^p and $\mu_m^{p^c}$ are the mean expression levels of the mth marker of cell type p and cell type p^c, respectively. As $G^{p \to p^c}$

is a stochastic process, we denote its realization as $g^{p \to p^c}$. We further introduce an $M \times 1$ random vector T_n^b composed of M independent random variables $T_{n,m}^b \sim Beta(\alpha_m^{b,p}, \beta_m^{b,p})$. During sampling, T_n^b will first be sampled and its realization t_n^b will be plugged into $g^{p \to p^c}$.

The batch effect to be added to the nth cell event whose cell type is p in the bth batch is $\Psi^{b,p} = I(X_n^b \succ 0) \circ (\psi^b 1 + \theta^{b,p})$, where $\circ$ is the Hadamard product, I is an indicator function, $\succ$ is a component-wise inequality, $\psi^b \sim N(0, \sigma_\psi^2)$, with $\sum_{b=1}^{B} \psi^b = 0$, 1 is an $M \times 1$ vector, $\theta_m^{b,p} \sim N(0, \sigma_\theta^2)$, with $\sum_{p=1}^{P} \theta_m^{b,p} = \sum_{m=1}^{M} \theta_m^{b,p} = 0$.

The temporal effect to be added to the nth cell event in the bth batch is $\Phi^b = I(X_n^b \succ 0) \circ h^b(\frac{n}{N^b})$, where $h^b(\frac{n}{N^b})$ is the realization of a stochastic process $H^b: [0,1]^M \mapsto R^M$, such that $H^b(0) = 0$.

With these setups, the simulated latent expression level for the nth cell event whose cell type is p and who is differentiating into cell type p^c in the bth batch is found by $Z_n^b = X_n^b + g^{p \to p^c}(T_n^p) + \Psi^{b,p} + \Phi^b$. The corresponding final expression level is $Y_n^b = I(Z_n^b \succ 0) \circ Z_n^b + E_n^b$, where $E_n^b \sim L$ and L is some error distribution.

Further details on the creation mode and the emulation mode

In this section, we provided some further details on the two simulation modes used in Cytomulate.

Creation mode: As no real data is required from the users, the initialization stage of the creation mode simply involves randomly generating average cell expressions as well as the GMMs to sample cell expressions from. If cell differentiation is to be simulated, the number of non-overlapping polytrees is needed as an extra input from the users. Cell types will then be randomly grouped and average cell expressions are randomly generated such that cells that are close to each other in a polytree will have similar expressions.

Emulation mode: The emulation mode works in a slightly different manner by exploiting the expression matrix and its associated cell type labels. Starting with the expressions of the pth cell type where $p = 1, \cdots, P$, Cytomulate will fit a GMM to its corresponding expression matrix and

use the resulting model as the model to sample cell expressions from. The Bayesian Information Criterion (BIC) is adopted for model selection. If simulating differentiation paths is sought after, the averages of the observed cell expressions of all cell types are used to construct a polytree or a polyforest.

Metrics

In this section, we elaborate on the metrics we used in the comparison studies.
Suppose in the simulated batch we have $m = 1, \cdots, M$ protein channels, N cell events, and $p = 1, \cdots, P$ cell types. We further assume that the cell abundance for the pth cell type is $n_p \geq 0$ such that $\sum_{p=1}^{P} n_p = N$ and that by y_n^p is the simulated $1 \times M$ cell expression vector for the nth cell event of the pth cell type. Then, for the pth cell type, we calculate the simulated mean as $m_p = \frac{1}{n_p} \sum_{n=1}^{n_p} y_n^p$ and the simulated covariance as $c_p = \frac{1}{n_p - 1} \sum_{n=1}^{n_p} (y_n^p)^T y_n^p$. We then carry out the same procedure on the observed dataset and find the corresponding m_p^r, and c_p^r. The L2 distance between the simulated mean and the observed mean is found via $dm_p = ||m_p - m_p^r||_2$. The Frobenius distance (54) between the simulated covariance and the observed covariance is calculated as $dc_p = ||c_p - c_p^r||_F$. And, the KL divergence dk_p is estimated following the method proposed in (41). Finally, the discrepancy measures for all the cell types are aggregated by a weighted average where the weights are decided by the cell abundances.

Code availability

The Cytomulate algorithm is available at cytomulate.readthedocs.io, and also shared through the Database of Actionable Immunology (dbai.biohpc.swmed.edu).

FUNDING SOURCES

This study was supported by the National Institutes of Health (NIH) [CCSG 5P30CA142543/TW, NIH 1R01CA258584/TW, XW, U01AI156189/TW], Cancer Prevention Research Institute of Texas [CPRIT RP190208/TW, XW], and Southern Methodist University [Dean's Research Council Award 2022/XW, TW].

AUTHOR CONTRIBUTIONS

Y.Y. performed all analyses. Y.Y. and K.W. created *Cytomulate* Package. Z.L provided LAMBDA analyses pipeline. X.W. conceived the study. T.W. and X.W. designed and supervised the whole study. All authors wrote the manuscript.

DECLARATION OF INTERESTS

The authors declare no competing interests related to this work.

Bibliography

1. Lou X, Zhang G, Herrera I, Kinach R, Ornatsky O, Baranov V, et al. Polymer-based elemental tags for sensitive bioassays. Angew Chem Int Ed. 2007;46(32):6111–4.
2. Spitzer MH, Nolan GP. Mass cytometry: single cells, many features. Cell. 2016 May 5;165(4):780–91.
3. Bandura DR, Baranov VI, Ornatsky OI, Antonov A, Kinach R, Lou X, et al. Mass cytometry: technique for real time single cell multitarget immunoassay based on inductively coupled plasma time-of-flight mass spectrometry. Anal Chem. 2009 Aug 15;81(16):6813–22.
4. Chester C, Maecker HT. Algorithmic Tools for Mining High-Dimensional Cytometry Data. J Immunol. 2015 Aug 1;195(3):773–9.
5. Finck R, Simonds EF, Jager A, Krishnaswamy S, Sachs K, Fantl W, et al. Normalization of mass cytometry data with bead standards. Cytometry A. 2013 May;83(5):483–94.
6. Trussart M, Teh CE, Tan T, Leong L, Gray DH, Speed TP. Removing unwanted variation with CytofRUV to integrate multiple CyTOF datasets. eLife. 2020 Sep 7;9.
7. Van Gassen S, Gaudilliere B, Angst MS, Saeys Y, Aghaeepour N. Cytonorm: A normalization algorithm for cytometry data. Cytometry A. 2020 Mar;97(3):268–78.
8. Shaham U, Stanton KP, Zhao J, Li H, Raddassi K, Montgomery R, et al. Removal of batch effects using distribution-matching residual networks. Bioinformatics. 2017 Aug 15;33(16):2539–46.
9. Upadhyay U, Jain A. Removal of Batch Effects using Generative Adversarial Networks. arXiv. 2019;
10. Van Gassen S, Callebaut B, Van Helden MJ, Lambrecht BN, Demeester P, Dhaene T, et al. FlowSOM: Using self-organizing maps for visualization and interpretation of cytometry data. Cytometry A. 2015 Jul;87(7):636–45.
11. Abe K, Minoura K, Maeda Y, Nishikawa H, Shimamura T. Model-based clustering for flow and mass cytometry data with clinical information. BMC Bioinformatics. 2020 Sep 17;21(Suppl 13):393.
12. Ji D, Nalisnick E, Qian Y, Scheuermann RH, Smyth P. Bayesian trees for automated cytometry data analysis. BioRxiv. 2018 Sep 19;
13. Dai Y, Xu A, Li J, Wu L, Yu S, Chen J, et al. CytoTree: an R/Bioconductor package for analysis

and visualization of flow and mass cytometry data. BMC Bioinformatics. 2021 Mar 22;22(1):138.

14. Wang K, Yang Y, Wu F, Song B, Wang X, Wang T. A Comparison of Dimension Reduction Methods for Cytometry by Time-of-Flight Data. BioRxiv. 2022 Apr 27;

15. Laurens van der M, Geoffrey H. Visualizing Data using t-SNE. J Mach Learn Res. 2008;

16. McInnes L, Healy J, Melville J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv. 2018;

17. Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition. Springer; 2008.

18. Amodio M, Srinivasan K, van Dijk D, Mohsen H, Yim K, Muhle R, et al. Exploring Single-Cell Data with Multitasking Deep Neural Networks. BioRxiv. 2017 Dec 19;

19. Ding J, Condon A, Shah SP. Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. Nat Commun. 2018 May 21;9(1):2002.

20. Hahne F, LeMeur N, Brinkman RR, Ellis B, Haaland P, Sarkar D, et al. flowCore: a Bioconductor package for high throughput flow cytometry. BMC Bioinformatics. 2009 Apr 9;10:106.

21. Hao Y, Hao S, Andersen-Nissen E, Mauck WM, Zheng S, Butler A, et al. Integrated analysis of multimodal single-cell data. Cell. 2021 Jun 24;184(13):3573-3587.e29.

22. Robinson MD, McCarthy DJ, Smyth GK. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics. 2010 Jan 1;26(1):139–40.

23. Lun ATL, Bach K, Marioni JC. Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. Genome Biol. 2016 Apr 27;17:75.

24. Vallejos CA, Marioni JC, Richardson S. BASiCS: Bayesian Analysis of Single-Cell Sequencing Data. PLoS Comput Biol. 2015 Jun 24;11(6):e1004333.

25. Zappia L, Phipson B, Oshlack A. Splatter: simulation of single-cell RNA sequencing data. Genome Biol. 2017 Sep 12;18(1):174.

26. Data scientist's primer to analysis of mass cytometry data [Internet]. [cited 2022 May 26]. Available from: https://biosurf.org/cytof_data_scientist.html

27. Levine JH, Simonds EF, Bendall SC, Davis KL, Amir ED, Tadmor MD, et al. Data-Driven Phenotypic Dissection of AML Reveals Progenitor-like Cells that Correlate with Prognosis. Cell. 2015 Jul 2;162(1):184–97.

28. Kaushik A, Dunham D, He Z, Manohar M, Desai M, Nadeau KC, et al. CyAnno: A semi-automated approach for cell type annotation of mass cytometry datasets. Bioinformatics. 2021 May 25;

29. Lukas M. CS. HDCytoData. Bioconductor. 2018;

30. Leipold MD, Obermoser G, Fenwick C, Kleinstuber K, Rashidi N, McNevin JP, et al. Comparison of CyTOF assays across sites: Results of a six-center pilot study. J Immunol Methods. 2018 Feb;453:37–43.

31. Street K, Risso D, Fletcher RB, Das D, Ngai J, Yosef N, et al. Slingshot: cell lineage and

pseudotime inference for single-cell transcriptomics. BMC Genomics. 2018 Jun 19;19(1):477.

32. Trapnell C, Cacchiarelli D, Grimsby J, Pokharel P, Li S, Morse M, et al. The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. Nat Biotechnol. 2014 Apr;32(4):381–6.

33. Saelens W, Cannoodt R, Todorov H, Saeys Y. A comparison of single-cell trajectory inference methods. Nat Biotechnol. 2019 May;37(5):547–54.

34. Bishop CM. Pattern Recognition and Machine Learning (Information Science and Statistics). Softcover reprint of the original 1st ed. 2006. Springer; 2016.

35. Liu X, Song W, Wong BY, Zhang T, Yu S, Lin GN, et al. A comparison framework and guideline of clustering methods for mass cytometry data. Genome Biol. 2019 Dec 23;20(1):297.

36. Weber LM, Robinson MD. Comparison of clustering methods for high-dimensional single-cell flow and mass cytometry data. Cytometry A. 2016 Dec 19;89(12):1084–96.

37. Lee H-C, Kosoy R, Becker CE, Dudley JT, Kidd BA. Automated cell type discovery and classification through knowledge transfer. Bioinformatics. 2017 Jun 1;33(11):1689–95.

38. Zhang Z, Luo D, Zhong X, Choi JH, Ma Y, Wang S, et al. SCINA: A Semi-Supervised Subtyping Algorithm of Single Cells and Bulk Samples. Genes (Basel). 2019 Jul 12;10(7).

39. Samusik N, Good Z, Spitzer MH, Davis KL, Nolan GP. Automated mapping of phenotype space with single-cell data. Nat Methods. 2016 Jun;13(6):493–6.

40. Hawley D, Ding J, Thotakura S, Haskett S, Aluri H, Kublin C, et al. RNA-Seq and CyTOF immuno-profiling of regenerating lacrimal glands identifies a novel subset of cells expressing muscle-related proteins. PLoS ONE. 2017 Jun 29;12(6):e0179385.

41. Perez-Cruz F. Kullback-Leibler divergence estimation of continuous distributions. 2008 IEEE International Symposium on Information Theory. IEEE; 2008. p. 1666–70.

42. Grimmett GR, Stirzaker DR. Probability and Random Processes. 3rd ed. Oxford: Oxford University Press; 2001.

43. Akima H. A new method of interpolation and smooth curve fitting based on local procedures. J ACM. 1970 Oct 1;17(4):589–602.

44. Dasgupta S. Learning Polytrees. arXiv. 2013;

45. Cormen TH, Leiserson CE, Rivest RL, Stein C. Introduction to Algorithms, 3rd Edition (The MIT Press). 3rd ed. Cambridge, Mass: The MIT Press; 2009.

46. Clauset A, Newman MEJ, Moore C. Finding community structure in very large networks. Phys Rev E. 2004 Dec 6;70(6).

47. Hagberg AA, Schult DA, Swart PJ. Exploring Network Structure, Dynamics, and Function using NetworkX. In: Varoquaux G, Vaught T, Millman J, editors. Proceedings of the 7th Python in Science conference. SciPy conference proceedings; 2008. p. 11–5.

48. Goodfellow I, Bengio Y, Courville A. Deep Learning (Adaptive Computation and Machine Learning series). Illustrated. Cambridge, Massachusetts: The MIT Press; 2016.

49. Ruthotto L, Haber E. An Introduction to Deep Generative Modeling. arXiv. 2021;

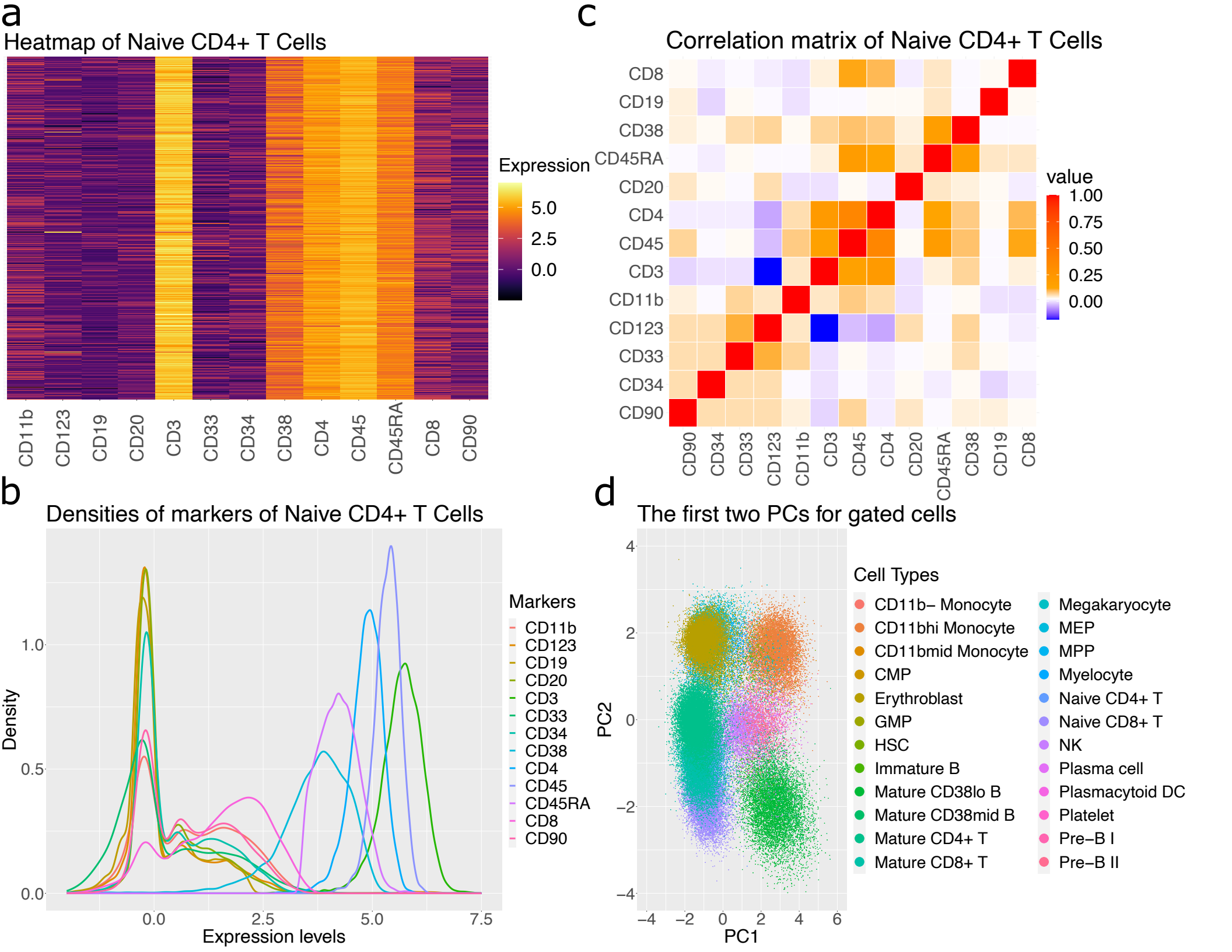
50. Bond-Taylor S, Leach A, Long Y, Willcocks CG. Deep Generative Modelling: A Comparative Review of VAEs, GANs, Normalizing Flows, Energy-Based and Autoregressive Models. arXiv. 2021;

51. Rezende DJ, Mohamed S. Variational Inference with Normalizing Flows. arXiv. 2015;

52. Kingma DP, Welling M. Auto-Encoding Variational Bayes. arXiv. 2013;

53. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative Adversarial Networks. arXiv. 2014;

54. Trefethen LN, David Bau III. Numerical Linear Algebra. 1st ed. Philadelphia: SIAM: Society for Industrial and Applied Mathematics; 1997.

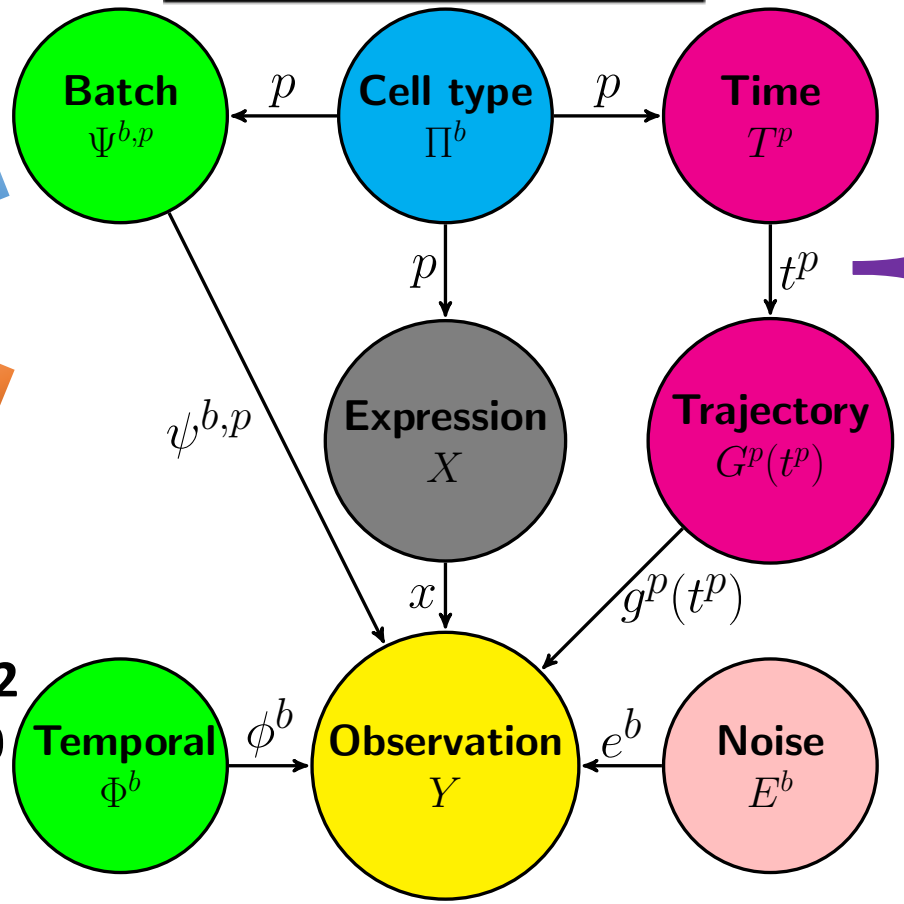

a
Heatmap of Naive CD4+ T Cells
Expression
5.0
2.5
0.0
CD11b
CD123
CD19
CD20
CD3
CD33
CD34
CD38
CD4
CD45
CD45RA
CD8
CD90
c
Correlation matrix of Naive CD4+ T Cells
value
1.00
0.75
0.50
0.25
0.00
CD8
CD19
CD38
CD45RA
CD20
CD4
CD45
CD3
CD11b
CD123
CD33
CD34
CD90
b
Densities of markers of Naive CD4+ T Cells
Density
1.0
0.5
0.0
0.0
2.5
5.0
7.5
Expression levels
Markers
CD11b
CD123
CD19
CD20
CD3
CD33
CD34
CD38
CD4
CD45
CD45RA
CD8
CD90
d
The first two PCs for gated cells
PC2
4
2
0
−2
−4
−4
−2
0
2
4
6
PC1
Cell Types
CD11b− Monocyte
CD11bhi Monocyte
CD11bmid Monocyte
CMP
Erythroblast
GMP
HSC
Immature B
Mature CD38lo B
Mature CD38mid B
Mature CD4+ T
Mature CD8+ T
Megakaryocyte
MEP
MPP
Myelocyte
Naive CD4+ T
Naive CD8+ T
NK
Plasma cell
Plasmacytoid DC
Platelet
Pre−B I
Pre−B II

$$\begin{pmatrix} 0 & \cdots & 2.72 \\ \vdots & \ddots & \vdots \\ 3.14 & \cdots & 0.58 \end{pmatrix}$$

$$\begin{pmatrix} \text{CD8 T cell} \\ \vdots \\ \text{B cell} \end{pmatrix}$$

# of batches: 3

Emulation Mode



Creation Mode

of cell types: 10

# of protein channels: 32

# of cell events: 100,000

⋮

# of batches: 3

$$\begin{pmatrix} 0 & \cdots & 2.72 \\ \vdots & \ddots & \vdots \\ 3.14 & \cdots & 0.58 \end{pmatrix}$$

$$\begin{pmatrix} \text{CD8 T cell} \\ \vdots \\ \text{B cell} \end{pmatrix}$$

Batch 1

$$\begin{pmatrix} 0 & \cdots & 2.72 \\ \vdots & \ddots & \vdots \\ 3.14 & \cdots & 0.58 \end{pmatrix}$$

$$\begin{pmatrix} \text{CD8 T cell} \\ \vdots \\ \text{B cell} \end{pmatrix}$$

Batch 2

$$\begin{pmatrix} 0 & \cdots & 2.72 \\ \vdots & \ddots & \vdots \\ 3.14 & \cdots & 0.58 \end{pmatrix}$$

$$\begin{pmatrix} \text{CD8 T cell} \\ \vdots \\ \text{B cell} \end{pmatrix}$$

Batch 3

Downstream Analysis

Dimension Reduction

+

Clustering

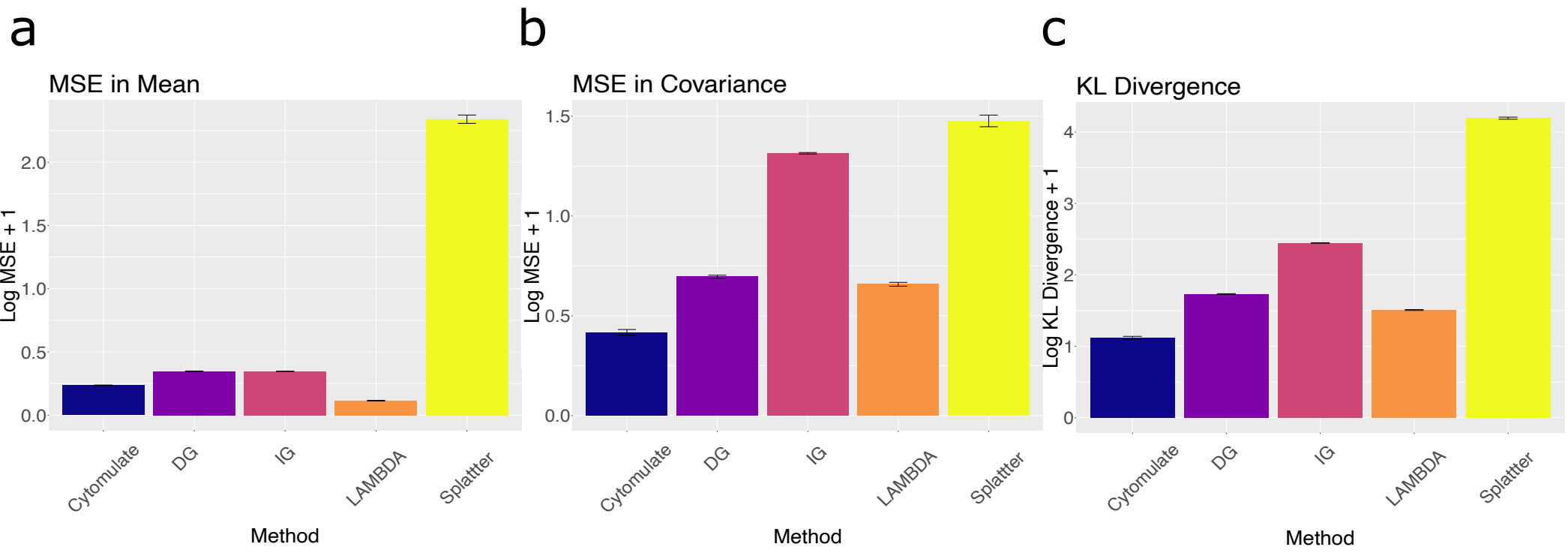
+

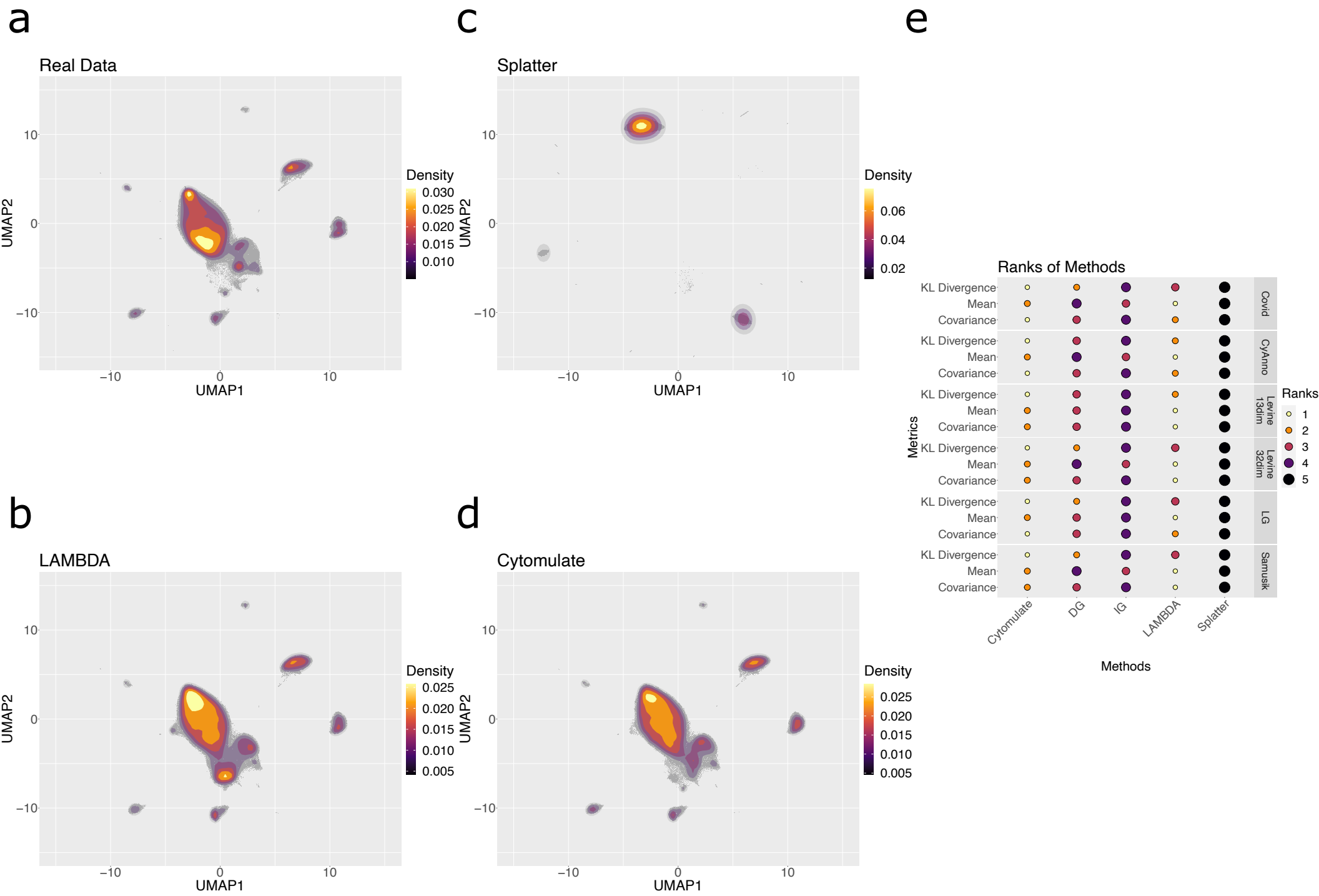
Batch Normalization

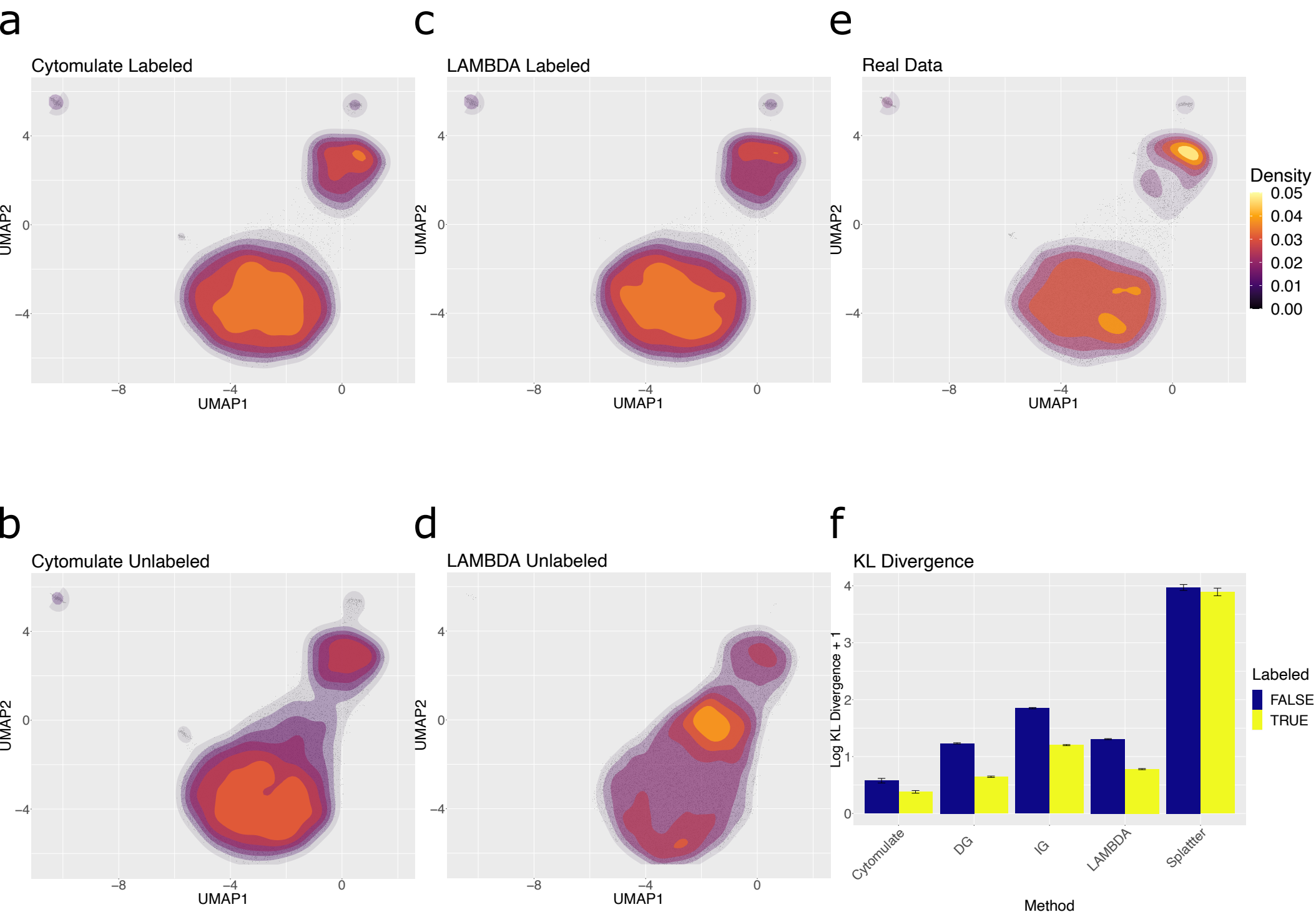
+

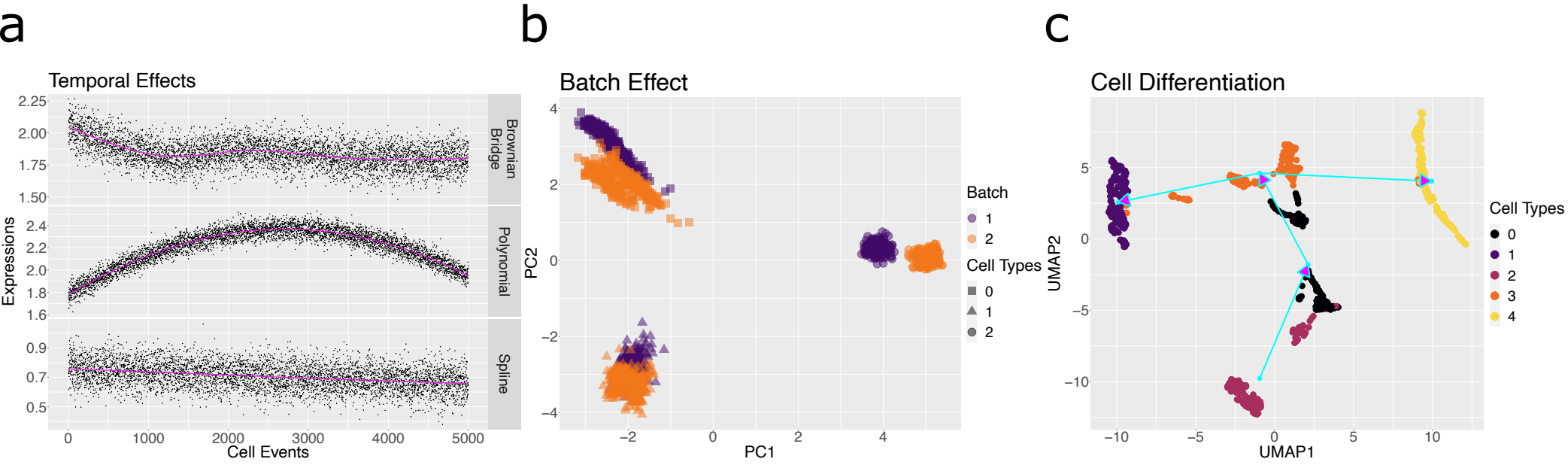
Trajectory Inference

⋮


a
MSE in Mean
Log MSE + 1
0.0
0.5
1.0
1.5
2.0
Cytomulate
DG
IG
LAMBDA
Splattter
Method
b
MSE in Covariance
Log MSE + 1
0.0
0.5
1.0
1.5
Cytomulate
DG
IG
LAMBDA
Splattter
Method
c
KL Divergence
Log KL Divergence + 1
0
1
2
3
4
Cytomulate
DG
IG
LAMBDA
Splattter
Method


a
Real Data
UMAP2
UMAP1
Density
0.030
0.025
0.020
0.015
0.010
c
Splatter
UMAP2
UMAP1
Density
0.06
0.04
0.02
e
Ranks of Methods
KL Divergence
Mean
Covariance
Covid
CyAmo
Levine 13dim
Levine 32dim
LG
Samusik
Metrics
Cytomulate
DG
IG
LAMBDA
Splatter
Methods
Ranks
1
2
3
4
5
b
LAMBDA
UMAP2
UMAP1
Density
0.025
0.020
0.015
0.010
0.005
d
Cytomulate
UMAP2
UMAP1
Density
0.025
0.020
0.015
0.010
0.005


a
Cytomulate Labeled
c
LAMBDA Labeled
e
Real Data
b
Cytomulate Unlabeled
d
LAMBDA Unlabeled
f
KL Divergence
UMAP1
UMAP2
Density
0.05
0.04
0.03
0.02
0.01
0.00
Log KL Divergence + 1
Method
Cytomulate
DG
IG
LAMBDA
Splatter
Labeled
FALSE
TRUE


a
Temporal Effects
Expressions
Cell Events
Brownian Bridge
Polynomial
Spline
b
Batch Effect
PC1
PC2
Batch
1
2
Cell Types
0
1
2
c
Cell Differentiation
UMAP1
UMAP2
Cell Types
0
1
2
3
4

Dataset	Species	Anatomic Site
Levine_32dim	Human	Bone Marrow
Levine_13dim	Human	Bone Marrow
Samusik	Mouse	Bone Marrow
CyAnno	Human	Peripheral Blood
Covid	Human	Peripheral Blood
LG	Mouse	Lacrimal Gland

Source	Number of Markers
HDCytoData	32
HDCytoData	13
HDCytoData	39
https://flowrepository.org/id/FR-FCM-Z2V9	39
https://dbai.biohpc.swmed.edu/cytof-database.php	43
https://flowrepository.org/experiments/1177/download_ziped_files	31

Number of Cells Events	Number of Cell Types
72,463	32
167,044	24
53,173	24
123,033	39
208,712	21
284,055	7