Supplementary information for **Differential cell-state abundance testing using k-NN graphs with *Milo***

Emma Dann, Neil C. Henderson, Sarah A. Teichmann, Michael D. Morgan, John C. Marioni

05 March, 2021

Contents

1 Supplementary notes

1.1 Description of workflow for *Milo* analysis

Given a single-cell dataset of gene expression profiles of M cells collected from S experimental samples, *Milo* aims to quantify systematic changes in abundance of cells between biological conditions, as compared to within-condition variability. Here we provide a step-by-step description of the workflow for differential abundance analysis. Of note, here we focus on application to single-cell gene expression profiles, and we provide guidelines for preprocessing on this type of data. However, the core of the *Milo* framework, from k-NN graph construction to differential abundance testing, is applicable to any kind of single-cell dataset that can be embedded in a low-dimensional space.

1.1.1 Preprocessing and dimensionality reduction

For preprocessing of scRNA-seq profiles we recommend following standard practices in single-cell analysis [1,2]: we normalize UMI counts by the total number of counts per cell, apply log-transformation and identify highly variable genes (HVGs). Then we project the $H \times M$ gene expression matrix, where M is the number of cells and H is the number of HVGs, to the first d principal components (PCs). While downstream analysis is generally robust to the exact choice of the number of HVGs [1], an optimal value for d can be selected by detecting the "elbow" in the variance explained by PCs or using the "jackstraw" method [3].

1.1.2 Minimizing batch effects

Comparing biological conditions often requires acquiring single-cell data from multiple samples, that can be generated with different experimental conditions or protocols. This commonly introduces batch effects, which can have a substantial impact on the data composition and subsequently the topology of any k-NN graph computed across the single-cell data. Consequently, this will have an impact on the ability of *Milo* to resolve genuine differential abundance of cells between experimental conditions of interest. In addition, other biological nuisance covariates could impact DA analysis i.e. biological factors that are not of interest for the analyst, such as donor of origin or sex of the donor. We recommend to mitigate the impact of technical or other nuisance covariates *before* building the k-NN graph, by using one of the many *in silico* integration tools designed for this task in single-cell datasets. Defining the best tool for this task is beyond the scope of this work, a large number of integration methods have been reviewed and benchmarked in [4–6]. However, users should consider the type of output produced by their integration method of choice, typically one of (A) a corrected feature space, (B) a joint embedding or (C) an integrated graph. The refined neighbourhood search procedure in *Milo* relies on finding neighbors in reduced dimension space. Therefore using a methods that produces an integrated graph (e.g. BBKNN [7], Conos [8]) could lead to suboptimal results in DA testing with *Milo*, because the refined neighbourhood search procedure would still be affected by the batch effect.

In addition, the effect of nuisance covariates should be modelled in the generalized linear model used for DA testing in *Milo* to minimize the emergence of false positives in case of imperfect batch correction (see Section 1.1.5) (Fig.2D).

We wish to emphasize that, when confounders are present, an appropriate experimental design is crucial to obtain reliable results from differential abundance analysis: if nuisance factors are 100% confounded with the biological condition used for differential abundance (e.g. if the samples from diseased and healthy donors are processed in separate sequencing batches), there is no way to disentangle the abundance differences that are truly driven by the biology of interest. We note that in a similar case applying a batch integration strategy before graph construction could lead to a loss of biological signal.

1.1.3 Building the k-NN graph

Milo uses a k-NN graph computed based on similarities in gene expression space as a representation of the phenotypic manifold in which cells lie. While *Milo* can be used on graphs built with different similarity kernels, here we compute the graph as follows: given the reduced dimension matrix X_{PC} of dimensions $M \times d$, for each cell j, the Euclidean distances to its K nearest neighbors in X_{PC} are computed and stored in a $M \times M$ adjacency matrix D. Then, D is made symmetrical, such that cells j and l are nearest neighbors (i.e. connected by an edge) if either j is a nearest neighbor of l or l is a nearest neighbor of j. The k-NN graph is encoded by the undirected symmetric version of $\tilde{D}$ of D, where each cell has at least K nearest neighbors.

1.1.4 Definition of cell neighbourhoods and index sampling algorithm

Next, we identify a set of representative cell neighbourhoods on the k-NN graph. We define the neighbourhood n_i of cell c_i as the group of cells that are connected to c_i by an edge in the graph. We refer to c_i

with $i = 1, 2, ..., N$ as the index cell of the neighbourhood, so that $N \leq M$. Formally, a cell c_j belongs to neighbourhood n_i if $\tilde{D}_{i,j} > 0$.

In order to define neighbourhoods that span the whole k-NN graph, we sample index cells by using an algorithm previously adopted for waypoint sampling for trajectory inference [9,10]. Briefly, we start by randomly sampling $p \cdot M$ cells from the dataset, where $p \in [0, 1]$ (we use $p = 0.1$ by default). Given the reduced dimension matrix used for graph construction X_{PC}, for each sampled cell c_j we consider its K nearest neighbors with PC profiles $x_1, x_2, ..., x_k$ and compute the mean position of the neighbors in PC space $\bar{x}$:

$$\bar{x_j} = \frac{\sum_k x_k}{K}$$

Then, we search for the cell c_i such that the Euclidean distance between x_i and $\bar{x}$ is minimized. Because the algorithm might converge to the same index cell from multiple initial samplings, this procedure yields a set of $N \leq p \cdot M$ index cells that are used to define neighbourhoods.

Having defined a set of N neighbourhoods from the sampled index cells, we construct a count matrix of dimensions $N \times S$ which reports, for each sample, the number of cells that are present in each neighbourhood.

1.1.5 Testing for differential abundance in neighbourhoods

To test for differential abundance between biological conditions, *Milo* models the cell counts in neighbourhoods, estimating variability across biological replicates using a generalized linear model (GLM). We build upon the framework for differential abundance testing implemented by *Cydar* [11]. In this section, we briefly describe the statistical model and adaptations to the k-NN graph setting.

Quasi-likelihood negative binomial generalized linear models We consider a neighbourhood n with cell counts y_{ns} for each experimental sample s. The counts are modelled by the negative binomial (NB) distribution, as it is supported over all non-negative integers and can accurately model both small and large cell counts. For such non-Normally distributed data we use generalized-linear models (GLMs) as an extension of classic linear models that can accomodate complex experimental designs. We therefore assume that

$$y_{ns} \sim NB(\mu_{ns}, \phi_n),$$

where μ_{ns} is the mean number of cells from sample s in neighbourhood n and ϕ_n is the NB dispersion parameter.

The expected count value μ_{ns} is given by

$$\mu_{ns} = \lambda_{ns} M_{ns}$$

where λ_{ns} is the proportion of cells belonging to experimental sample s in n and M_s is the total number of cells of s. In practice, λ_{ns} represents the biological variability that can be affected by treatment condition, age or any biological covariate of interest.

We use a log-linear model to model the influence of a biological condition on the expected counts in the neighbourhood:

$$log\ \mu_{ns} = \sum_{g=1}^{G} x_{sg}\beta_{ng} + log\ M_s$$

Here, for each possible value g taken by the biological condition of interest, x_{sg} is the binary vector indicating the condition value applied to sample s. β_{ng} is the regression coefficient by which the covariate effects are mediated for neighbourhood n, that represents the log fold-change between number of cells in condition g and all other conditions. If the biological condition of interest is ordinal (such as age or disease-severity) β_{ng} is interpreted as the per-unit linear change in neighbourhood abundance.

Estimation of β_{ng} for each n and g is performed by fitting the GLM to the count data for each neighbourhood, i.e. by estimating the dispersion ϕ_n that models the variability of cell counts for replicate samples for each neighbourhood. Dispersion estimation is performed using the quasi-likelihood method in `edgeR`[12], where the dispersion is modelled from the GLM deviance and thereby stabilized with empirical Bayes shrinkage, to stabilize the estimates in the presence of limited replication.

Adaptation of Spatial FDR to neighbourhoods WeTo account for the non-independence of spatially overlapping neighbourhoods by applying a weighted version of the Benjamini-Hochberg (BH) method, where P values are weighted by the reciprocal of the neighbourhood connectivity, we build upon as an adaptation to graphs of a previously described strategy to control the spatial False Discovery Rate (FDR) [6].

To control for multiple testing, we need to account for the overlap between neighbourhoods, that makes the differential abundance tests non-independent. We apply a weighted version of the Benjamini-Hochberg (BH) method, where P values are weighted by the reciprocal of the neighbourhood connectivity, as an adaptation to graphs of the Spatial FDR method introduced by *Cydar* [11]. Formally, to control for FDR at a selected threshold α we reject null hypothesis i where the associated p-value is less than the threshold:

$$\max_i p_{(i)} : p_{(i)} \leq \alpha \frac{\sum_{l=1}^{i} w_{(l)}}{\sum_{l=1}^{n} w_{(l)}}$$

Where the weight $w_{(i)}$ is the reciprocal of the neighbourhood connectivity c_i. As a measure of neighbourhood connectivity, we use the Euclidean distance between the neighbourhood index cell c_i and its kth nearest neighbour in PC space.

1.2 Guidelines on parameter choice

In this section we provide practical guidelines to select default parameters for k-NN graph and neighbourhood construction for DA analysis with Milo. We recognize results will also be impacted by choices made during feature selection and dimensionality reduction. However these depend strongly on the nature of the single-cell dataset used as input. For example feature selection strategies suitable for UMI-based scRNA-seq data might be suboptimal for data generated with non-UMI protocols, or dimensionality reduction methods alternative to PCA might be used for single-cell epigenomics data. We point the reader to existing resources and heuristics for the application to scRNA-seq in section 1.1.1.

Selecting the number of nearest neighbors K For construction of the k-NN graph and neighbourhoods, the user has to select the number of nearest neighbors K to use for graph construction. The choice of K influences the distribution of cell counts within neighbourhoods, as K represents the lower limit in the number of cells in each neighbourhood ($\sum(y_{n,s})$). Hence, if K is too small the neighbourhoods might not contain enough cells to detect differential abundance. On the other hand, increasing K too much might lead to "oversmoothing" of neighbourhoods, with the inability to detect changes in abundance. We illustrate this trade-off by testing for DA setting increasing values for K in the mouse gastrulation dataset with synthetic condition labels on 4 different populations (Suppl.Fig. . . .A).

In order to perform DA testing with sufficient statistical power, the analyst should consider the number of experimental samples S (that will correspond to the columns in the count matrix for DA testing) and the desired minimum number of cells per neighbourhood and experimental sample. For example, having on average 5 cells per sample in each neighbourhood allows to detect n fold changes between 2 experimental conditions. The median number of cells per sample in each neighbourhood $\hat{y}_{ns}$ increases with the total neighbourhood size (Suppl.Fig. . . .B), with:

$$\hat{y}_{ns} \sim \frac{\sum_s y_{ns}}{S}$$

Therefore a conservative approach is to select $K \geq (S\times$ 3-5. As the average neighbouhood size increases with K, we recommend to pick K by inspecting the histogram of neighbourhood sizes after sampling of neighbourhoods (Suppl.Fig.…C). In order to have at least 3-5 cells per sample, all the neighbourhoods should be bigger than $S \times 3 - 5$. We provide the utility function `plotNhoodSizeHist` to visualize the neighbourhood size distribution as part of our R package.

Selecting the proportions of cells sampled as neighbourhood indices p

References

1. Luecken, M.D., and Theis, F.J. (2019). Current best practices in single-cell RNA-seq analysis: A tutorial. Molecular Systems Biology *15*, e8746.

2. Amezquita, R.A., Lun, A.T.L., Becht, E., Carey, V.J., Carpp, L.N., Geistlinger, L., Marini, F., Rue-Albrecht, K., Risso, D., and Soneson, C. *et al.* (2020). Orchestrating single-cell analysis with Bioconductor. Nature Methods *17*, 137–145.

3. Chung, N.C., and Storey, J.D. (2015). Statistical significance of variables driving systematic variation in high-dimensional data. Bioinformatics *31*, 545–554.

4. Luecken, M.D., Büttner, M., Chaichoompu, K., Danese, A., Interlandi, M., Mueller, M.F., Strobl, D.C., Zappia, L., Dugas, M., and Colomé-Tatché, M. *et al.* (2020). Benchmarking atlas-level data integration in single-cell genomics. bioRxiv, 2020.05.22.111161.

5. Chazarra-Gil, R., Dongen, S. van, Kiselev, V.Y., and Hemberg, M. (2020). Flexible comparison of batch correction methods for single-cell RNA-seq using BatchBench. bioRxiv, 2020.05.22.111211.

6. Tran, H.T.N., Ang, K.S., Chevrier, M., Zhang, X., Lee, N.Y.S., Goh, M., and Chen, J. (2020). A benchmark of batch-effect correction methods for single-cell RNA sequencing data. Genome Biology *21*, 12.

7. Polański, K., Young, M.D., Miao, Z., Meyer, K.B., Teichmann, S.A., and Park, J.-E. BBKNN: Fast batch alignment of single cell transcriptomes. Bioinformatics.

8. Barkas, N., Petukhov, V., Nikolaeva, D., Lozinsky, Y., Demharter, S., Khodosevich, K., and Kharchenko, P.V. (2019). Joint analysis of heterogeneous single-cell RNA-seq dataset collections. Nat Methods *16*, 695–698.

9. Gut, G., Tadmor, M.D., Pe'er, D., Pelkmans, L., and Liberali, P. (2015). Trajectories of cell-cycle progression from fixed cell populations. Nature Methods *12*, 951–954.

10. Setty, M., Tadmor, M.D., Reich-Zeliger, S., Angel, O., Salame, T.M., Kathail, P., Choi, K., Bendall, S., Friedman, N., and Pe'er, D. (2016). Wishbone identifies bifurcating developmental trajectories from single-cell data. Nature Biotechnology *34*, 637–645.

11. Lun, A.T.L., Richard, A.C., and Marioni, J.C. (2017). Testing for differential abundance in mass cytometry data. Nature Methods *14*, 707–709.

12. Robinson, M.D., McCarthy, D.J., and Smyth, G.K. (2010). edgeR: A Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics *26*, 139–140.