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

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

John C. Marioni

18 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 KNN 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 KNN 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 KNN 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

43   work, a large number of integration methods have been reviewed and benchmarked in [4–6]. However, users

44   should consider the type of output produced by their integration method of choice, typically one of (A) a

45   corrected feature space, (B) a joint embedding or (C) an integrated graph. The refined neighbourhood search

46   procedure in *Milo* relies on finding neighbors in reduced dimension space. Therefore using a methods that

47   produces an integrated graph (e.g. BBKNN [7], Conos [8]) could lead to suboptimal results in DA testing

48   with *Milo*, because the refined neighbourhood search procedure would still be affected by the batch effect.

49   In addition, the effect of nuisance covariates should be modelled in the generalized linear model used for DA

50   testing in *Milo* to minimize the emergence of false positives in case of imperfect batch correction (see Section

51   1.1.5) (Fig.2D).

52   We wish to emphasize that, when confounders are present, an appropriate experimental design is crucial to

53   obtain reliable results from differential abundance analysis: if nuisance factors are 100% confounded with

54   the biological condition used for differential abundance (e.g. if the samples from diseased and healthy donors

55   are processed in separate sequencing batches), there is no way to disentangle the abundance differences that

56   are truly driven by the biology of interest. In a similar case applying a batch integration strategy before

57   graph construction could lead to a loss of biological signal.

58   ### 1.1.3   Building the KNN graph

59   *Milo* uses a KNN graph computed based on similarities in gene expression space as a representation of the

60   phenotypic manifold in which cells lie. While *Milo* can be used on graphs built with different similarity

61   kernels, here we compute the graph as follows: given the reduced dimension matrix X_{PC} of dimensions

62    $M \times d$, for each cell j, the Euclidean distances to its K nearest neighbors in X_{PC} are computed and stored in

63   a $M \times M$ adjacency matrix D. Then, D is made symmetrical, such that cells c_i and c_j are nearest neighbors

64   (i.e. connected by an edge) if either c_i is a nearest neighbor of c_j or c_j is a nearest neighbor of c_i. The KNN

65   graph is encoded by the undirected symmetric version of $\tilde{D}$ of D, where each cell has at least K nearest

66   neighbors.

67   ### 1.1.4   Definition of cell neighbourhoods and index sampling algorithm

68   Next, we identify a set of representative cell neighbourhoods on the KNN graph. We define the neighbourhood

69   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 KNN 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 KNN 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

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

where λ_{ns} is the proportion of cells belonging to experimental sample s in n and M_s is the sum of counts of cells of s over all the neighbourhoods. 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   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:

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

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 KNN graph and neighbourhood construction for DA analysis with Milo. We recognize that DA analysis 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 KNN 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. 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. . . .A), with:

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

Therefore a conservative approach to minimize false positives is to select $K \geq S \times$ 3-5.

On the other hand increasing K increases power, but can come at the cost of FDR control, as we illustrate by testing for DA with increasing values for K in the mouse gastrulation dataset with synthetic condition labels on 4 different populations (Suppl.Fig. . . .A). We recommend to inspect the histogram of neighbourhood sizes after sampling of neighbourhoods (Suppl.Fig. . . .B) and to consider what is the number of cells that would be considered a "neighbourhood" in the dataset at hand. Depending on the expected lower-limit of the size

6

of rare sub-populations, the mean neighbourhood size should be no more than 10% of this value. 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 The proportion of cells sampled for search of neighbourhood indices can affect the total number of neighbourhoods used for analysis, but this number will converge for high proportions thanks to the sampling refinement step described in section 1.1.4 (Suppl.Fig. 1A). In practice, we recommend initiating neighbourhood search with $p = 0.05$ for datasets with more than 100k cells and $p = 0.1$ otherwise, which we have found to give appropriate coverage across the KNN graph while reducing the computational and multiple-testing burden. We recommend increasing $p > 0.1$ only if the dataset appears to contain rare disconnected subpopulations.

1.3   On compositional biases

A common problem faced by statistical methods detecting shifts in abundance in single-cell datasets is to account for compositional biases: because the total number of cells per sample is relatively small, an increase in cell abundance in a single cell population can lead to an to a reduction in the number of cells in other populations. Therefore a statistical model that tests for changes in abundance *indipendently* for different cell populations might detect false DA simply because of shifts affecting a different population. The challenge of compositional dependencies has been addressed in the field of microbiome analysis [refs] and more recently for analysis of cell type shifts in single-cell RNA-seq datasets [13].

Reassuringly, although *Milo* does not model the compositional dependencies explicitly, we do not observe any of such biases across our benchmarking datasets: here we introduce a synthetic enrichment in one of two conditions, but we do not detect false depletion in other non-DA regions of the graph. We speculate that this problem is alleviated by performing tests over a large number of neighbourhoods (in the order of thousands)

References

168 1. Luecken, M.D., and Theis, F.J. (2019). Current best practices in single-cell RNA-seq analysis: A tutorial.

169 Molecular Systems Biology *15*, e8746.

170 2. Amezquita, R.A., Lun, A.T.L., Becht, E., Carey, V.J., Carpp, L.N., Geistlinger, L., Marini, F., Rue-

171 Albrecht, K., Risso, D., and Soneson, C. *et al.* (2020). Orchestrating single-cell analysis with Bioconductor.

172 Nature Methods *17*, 137–145.

173 3. Chung, N.C., and Storey, J.D. (2015). Statistical significance of variables driving systematic variation in

174 high-dimensional data. Bioinformatics *31*, 545–554.

175 4. Luecken, M.D., Büttner, M., Chaichoompu, K., Danese, A., Interlandi, M., Mueller, M.F., Strobl, D.C.,

176 Zappia, L., Dugas, M., and Colomé-Tatché, M. *et al.* (2020). Benchmarking atlas-level data integration in

177 single-cell genomics. bioRxiv, 2020.05.22.111161.

178 5. Chazarra-Gil, R., Dongen, S. van, Kiselev, V.Y., and Hemberg, M. (2020). Flexible comparison of batch

179 correction methods for single-cell RNA-seq using BatchBench. bioRxiv, 2020.05.22.111211.

180 6. Tran, H.T.N., Ang, K.S., Chevrier, M., Zhang, X., Lee, N.Y.S., Goh, M., and Chen, J. (2020). A

181 benchmark of batch-effect correction methods for single-cell RNA sequencing data. Genome Biology *21*, 12.

182 7. Polański, K., Young, M.D., Miao, Z., Meyer, K.B., Teichmann, S.A., and Park, J.-E. BBKNN: Fast batch

183 alignment of single cell transcriptomes. Bioinformatics.

184 8. Barkas, N., Petukhov, V., Nikolaeva, D., Lozinsky, Y., Demharter, S., Khodosevich, K., and Kharchenko,

185 P.V. (2019). Joint analysis of heterogeneous single-cell RNA-seq dataset collections. Nat Methods *16*,

186 695–698.

187 9. Gut, G., Tadmor, M.D., Pe'er, D., Pelkmans, L., and Liberali, P. (2015). Trajectories of cell-cycle

188 progression from fixed cell populations. Nature Methods *12*, 951–954.

189 10. Setty, M., Tadmor, M.D., Reich-Zeliger, S., Angel, O., Salame, T.M., Kathail, P., Choi, K., Bendall,

190 S., Friedman, N., and Pe'er, D. (2016). Wishbone identifies bifurcating developmental trajectories from

191 single-cell data. Nature Biotechnology *34*, 637–645.

192 11. Lun, A.T.L., Richard, A.C., and Marioni, J.C. (2017). Testing for differential abundance in mass

193 cytometry data. Nature Methods *14*, 707–709.

194 12. Robinson, M.D., McCarthy, D.J., and Smyth, G.K. (2010). edgeR: A Bioconductor package for differ-

195 ential expression analysis of digital gene expression data. Bioinformatics *26*, 139–140.

196 13. Büttner, M., Ostner, J., Müller, C., Theis, F., and Schubert, B. (2020). scCODA: A Bayesian model for

197 compositional single-cell data analysis. bioRxiv.