Appendix S1: MetaCell - Correcting and clustering single cell RNA-seq data using k-nn graph covering

Summary

Single cell RNA-seq (scRNA-seq) experiments probe the distributions of cellular mRNA in complex cell populations by implementing massively parallel schemes for sampling cells from tissues and molecules from cells. We describe a general computational methodology for analyzing scRNA-seq datasets by covering single cells cohorts with metacells, which are constructed as coherent and homogeneous groups of cells for use in downstream analysis. We show how to apply metacell covers for identification and removal of batch effects, ambient noise, and doublets. We also develop a bootstrap strategy for detecting robust clusters in the data through resampling and metacell cover, and a technique to derive 2D projections of scRNA data by drawing the metacells graph and overlaying cells over it. Metacells thereby facilitate a non-parametric pre-processing of scRNA-seq datasets, simplifying statistical analyses and downstream modeling of complex regulatory phenomena. We envisage that this approach will become increasingly effective as datasets scale from hundreds to millions of cells and metacells become increasingly precise.

INTRODUCTION

Single cell RNA-seq (scRNA-seq) is facilitating massively parallel acquisition of transcriptional profiles from heterogeneous cell populations. The derived cohorts of single cell profiles are analyzed in order to detect cell types, cell sub-types and continuous gene expression gradients. These phenomenological observations may be linked with different types of dynamics, including development and differentiation, cell cycle, response to stimuli and more[1–8] (reviewed in [9]). A key challenge in the analysis of scRNA-seq data is the discrete, sparse and variable nature of the cellular mRNA molecule census. In mammals, a total of 10^4-10^6 copies of mRNA represent over 20,000 potential transcripts and these transcripts show over four orders of magnitude variation in abundance between highly expressed (tens of thousands of molecules per cell) and lowly expressed (less than 1 molecule per cell) genes. The

scRNA-seq experimental procedure further samples these mRNA distributions to provide typically 10,000 unique molecule identifiers (UMI) per cell, but less than 1000 molecules for many important populations of small cells. Moreover, even mRNA species that are sampled less than 0.1 times on average per cell (within a certain cell population) may represent an overall transcriptional output that sustains functional levels of key proteins. scRNA-seq analysis is therefore fundamentally different from classical gene expression analysis, and relies on inference of molecular behaviors based on a large number of partial observations rather than direct comparison of (perhaps noisy or biased, but otherwise comprehensive) profiles.

The combination of discrete and sparse scRNA profiles, a poorly characterized biological distribution of transcription states in single cells, and multiple sources of experimental error and bias are together challenging the classical serial pipeline approach for analyzing gene expression data. In particular, it is difficult to implement effective filtering of low-level sources of technical noise in order to produce data with guaranteed high quality for downstream high-level analysis (identify clusters, infer dynamics, or test gene regulatory hypotheses). Conversely, it is difficult to analyze the results of these high-level analyses without revisiting potential sources of low-level noise in the data. A series of computational and experimental advances were introduced to meet those challenges in recent years[10–22], but effective and robust analysis of complex scRNA-seq data still involve substantial efforts by both computational and domain experts.

Here we report on a set of computational tools that we developed to study complex scRNA-seq data. Our approach includes procedures for selecting informative genes, filtering background noise and outliers, clustering, and visualization. At the core of all these procedures are algorithms for covering the single cell dataset with metacells. A metacell is defined as a group of cells that are similar to each other given some simplified parametric hypothesis on cellular RNA distribution (e.g. assuming molecule counts are sampled from a multinomial sample), and that for a dense subgraph in a K-nn regularized similarity graph constructed over all cells. Each metacell pools together data from dozens to hundreds of cells, thereby allowing for accurate intra-metacell inference of expression. Metacells can therefore be viewed as an approximation for the entire dataset. Locally, the expression within each metacell is modeled using a multinomial or other simple family of distributions that assume (precise or approximated) conditional independence among genes. Globally, the complex expression landscape is piecewise approximated in a non-parametric

fashion where cells are assumed conditionally independent given their metacell associations. Using metacells as approximations for the data, we can derive simple strategies for handling noise and bias. We can also use metacells as a basis for deriving cell clusters and assessing their robustness.

Most of the ideas we introduce and use below are present in some of the recent literature on scRNA-seq analysis, but we believe that the details of our implementation and, in particular, the native handling of scRNA-seq data as multinomial samples from limited RNA pool are advantageous in practice. We demonstrate the computational principles discussed below by analyzing a data set including 8,500 peripheral blood cells (10X genomics) which represent a mixture of several blood cell types and was used before to illustrate scRNA-seq analysis.

METHODS AND RESULTS

Notation and pre-processing. We assume raw reads are mapped to genome sequences and assigned to cell barcodes and unique molecular identifiers (UMI) using robust pipelines that eliminate most of the UMI duplications induced by PCR and sequencing errors. We summarize all UMIs in the molecule count matrix $U = [u_{gi}]$ on genes $g \in G$ and cells $i \in I$. Each cell may be associated with a batch identifier, that we represent using a vector b_i over the cells.

We assume a set of gene features $\mathrm{F} \subseteq \mathrm{G}$ is specified and focus our analysis of some normalized form of the features for each cell. This allow very direct statistical and biological interpretation of the molecule distribution per gene feature, while of course requiring us to work over a feature space in high dimension.

For simplicity, we use the following conventions on matrices: Given a matrix $A = [a_{ij}]$ we denote marginalization over rows as a_i and columns as a_j. For example, u_g is the total molecule count for gene g on the raw count matrix, and u_i is the total number of molecule for a cell (sometime referred to as the cell's *depth*, or the cell's *complexity*). To define matrix multiplication, we use implicit summation over shared indices $AB = [a_{ij} + b_{jk}]$. We also freely employ matrix subsetting. Given any subset of indices I' in I, we denote by $A[I',] = [a_{i'j}]_{i' \in I'}$.

The procedures below are designed to robustly define a metacell structure while filtering outliers and noisy cells, and we are therefore not assuming the count matrix to be completely devoid of problematic profiles (e.g. as those originating from empty wells or droplets). Nevertheless, we implicitly assume that the fraction of completely

noisy profiles is relatively low, and some initial threshold of minimal cell UMI count must be employed.

Minimal cell filtering is applied at this initial phase. Cells representing empty wells/droplets, noisy wells/droplets or doublets are further analyzed and corrected downstream, based on richer models of the multivariate single cells gene expression distributions obtained once initial metacells are derived. Similarly, explicit normalization of cell depth is deferred to later stages, to allow downstream consideration of cell types with different depth distributions and minimize potential biases associated with early normalization.

Selection of feature genes. The data matrix U contains a large number of genes with variable intensities (typically spanning 3-4 orders of magnitude) and degree of cell-to-cell heterogeneity. We select genes for modeling cell-to-cell similarity using a combination of several approaches that allow the analyst to determine a suitable strategy given the biological question at hand:

- *Normalized variance* – we define a down-sampling threshold T_{ds} using the 10th percentile of the u_i distribution. We then generate a normalized matrix $W = [w_{gi}]$ by sampling from the list of molecules defined by each column in $U[, u_i > T_{ds}]$ exactly T_{ds} molecules without replacement. Rows in the matrix W are defined by their mean UMI count e_g^{ds} and by their variance v_g^{ds}. The variance is expected to be affected by three components. First, since molecules are being sampled from each cell, the sampling variance is expected to be in the order of the mean number of sampled molecules, with a distribution that follows a binomial model (*sampling variance*). Second, RNA concentrations of a gene within homogeneous cell populations are subject to stochastic control that contributes additional variance to our sample (*stochastic variance*). Third, when observing heterogeneous single cell populations, genes that are differentially expressed will show additional strong variance

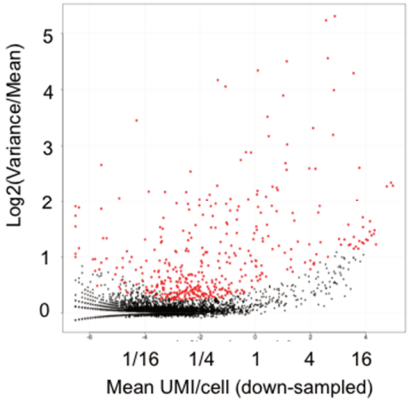

Figure 1. Gene selection using normalized variance. For each gene, we compare the mean UMIs per cell in the down-sampled PBMC 8k dataset, to the log ratio between variance and mean. Genes with variance that is higher than a non-parametric trend line are marked in red, and selected as features for downstream analysis. For highly expressed genes, the variance is affected by a noticeable component of stochastic gene regulation, and it scales faster than the mean, requiring an empirical correction rather than a simple threshold on the normalized variance.

associated with the sampled sub-populations or cell types (*regulatory variance*). The variance in low expression genes ($e_g^{ds} < 1$) will be dominated by the sampling component, but for genes with higher variance it is difficult to separate stochastic from regulatory variance (in fact, stochastic variance can even be regulated in a cell-type specific fashion). We therefore choose to remove the sampling variance and assume in this context that the residual variance is entirely regulatory. We first normalize the sampling variance $v'_g = log2\left(\frac{v_g^{ds}}{e_g^{ds}}\right)$ and then compute the empirical trend $v'(e)$ using a moving median with a window of 100 genes (using the median of the left-most and right most 100 genes for the edges). Finally, we recalibrate variance over this trend as $v_g'^{ds} = v_g^{ds} - v'(e_g)$. Genes with $v_g'^{ds} > T_{vm}$ are selected as features, where T_{vm} = 0.1 by default (**Figure 1**).

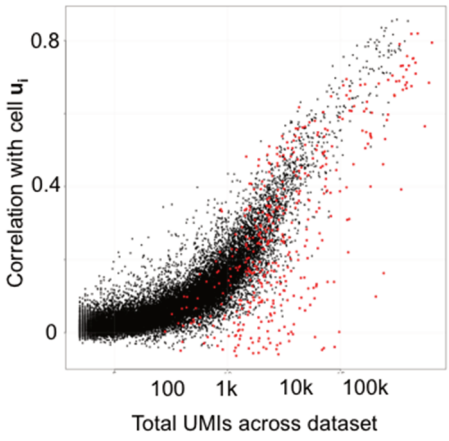

Figure 2. Gene selection using normalized depth scaling. For each gene, we compare the total UMIs (in log scale) to the Pearson correlation between the gene UMI count vector per cell and the total number of UMIs per cell (cell depth). Genes that are selected by the normalized variance scheme (Figure 1) are highlighted in red. Alternatively, we can select genes based on the reduction in size correlation compared to the global trend depicted here. Note the imperfect consistency between gene selection using variance and using depth correlation. The main cause for discrepancies is statistical linkage between cell size and UMI count, and between cell size and cell type, which make the depth scaling gene selection scheme biased against genes marking large cell types. Selection using variance is however biased against small cell types, since molecule down-sampling must be used for balancing the sampling variance across different single cells.

- *Normalized depth scaling*. Computing normalized variance forces us to down-sample the matrix, which is problematic in cases where the $\boldsymbol{u_i}$ distribution is highly variable itself. Indeed, in many cases this cell depth distribution can range from a few hundreds to over 20,000 molecules. In such cases, we can compute for each gene g the Pearson correlation with the cell depth $r_g^{sz} = cor(u_{gi}, u_i)$. As genes increase in their expression, their correlation with the cell depth will increase, even if their expression is completely homogeneous within the population, simply due to the decreasing sampling variance of their concentrations. On the other hand, truly variable genes will show a lower correlation with the cell depth compared to genes with similar expression average. We therefore compute an empirical trend $r(u_g)$ using the median $\boldsymbol{r_g^{sz}}$ correlation in a moving window of total gene expression, using a window of 100 genes. Then, we define the normalized depth scaling as $r'_g = r_g^{sz} - r(u_g)$. Finally, we select genes with sufficiently high u_g and $r'_g < T_{gr}$ (T_{gr} = -0.1 by

default). We note that there are datasets in which specific biological populations are characterized by larger cells and higher depth, which may result in high r' values for important genes. The analyst must make a decision on how to balance gene selection using variance and depth scaling depending on the biological application (**Figure 2**).

- *Batch features elimination*. Optionally, we can filter genes that are differentially expressed between batches at the initial feature selection phase. One approach for detecting such genes is to perform chi square test on the table of total UMI count per gene per batch, and total UMI count per all genes per batch and manually review the list of genes detected as biased. It should be noted that in many applications, such early batch correction is inappropriate, since one cannot assume batches are reproducing the same cell type compositions precisely. We therefore recommend addressing batch biases in later stages of the algorithm (see below).

- *Blacklisting*. We can define a list of genes (or putative transcripts) that are excluded from the candidate gene features, based on prior knowledge associating them with technical noise, or biological processes that are not at the primary focus of the analysis (e.g. stress or cell cycle).

The above metrics allow us to identify a subset of the genes F that will subsequently be used to define similarities between cells and group them into metacells. We suggest that the feature selection process should be optimized and adapted to the biological question at hand, since the distinction between “interesting” and “less interesting” sources of variation in the data is hard to define without a biological context.

Feature normalization and raw similarity matrix. We transform the raw UMI count U on the gene features F as $M = \left[m_{gi}\right] = \left[log_2\left(\epsilon + u_{gi}\right)\right]_{g\in F}$. The parameter ϵ is discussed below. We then compute the raw similarity matrix using Pearson correlations on the transformed features $R = \left[r\left(m_{gi}, m_{gj}\right)\right]_{ij}$. We motivate this approach as follows: Two similar cells are ideally generated from the same probabilistic model that is defined by a log-normal distribution of concentrations for each gene. For simplicity we assume constant variance and variable mean for all genes. To sample single cell profiles from this model, we first draw from the log-normal distributions to generate a multinomial parameters for each cell, and then

sample u_i and u_j molecules from these models respectively. Conversely, given data u_{gi} the Bayesian estimate for the multinomial parameter with a uniform prior is $u_{gi} + \epsilon$, and the log likelihood of $u_{gj} + \epsilon$ given the multinomial estimate on I and a log-normal model with constant variance is up to a constant $\sum_g (\log(u_{gi} + \epsilon) - \log(u_{gj} + \epsilon))^2$. To determine the prior, we note that the posterior estimate of the mean expression level of a gene given 0, 1 or 2 sampled molecules will scale like $\log(\epsilon), \log(\epsilon + 1), \log(\epsilon + 2)$ suggesting calibration of ϵ such that $\log(\epsilon + 1) - \log(\epsilon)$ and $\log(\epsilon + 2) - \log(\epsilon + 1)$ expresses our belief on the expected fold change in gene expression when sampling zero UMI to one UMI, vs. case comparing two UMIs and one UMI.

We note that a variation on the definition of M and R may involve normalizing the U matrix to fixed number of molecules per cell through sampling without replacement. When avoiding this, it is essential to employ alternative mechanisms that can compensate for the higher correlation between cells that are sampled more deeply as discussed below.

The metacell balanced K-nn cell similarity graph. Next, based on the raw similarity matrix R we perform a non-parametric transformation and compute $S = [s_{ij}] = [rank_j(r_{ij})]$. Here $rank$ is the ranking function, and each row represents the order of similarity between all cells j and a specific cell i. The S matrix is highly non-symmetric, in particular when the similarities going from an outlier cell are linking it to members of a large and homogeneous cell group. In such cases, the outlier cell will not be among the most similar cells to any of its own neighbors. To better consider such effects, we symmetrize S and balance the resulting matrix through the following steps:

$$[s^1_{ij}] = [max(\alpha K^2 - s_{ij} * s_{ji}, 0)]$$

$$[s^2_{ij}] = [max(\beta K - rank_j(s^1_{ij}), 0)]$$

$$[a_{ij}] = [max(K - rank_j(s^2_{ij}), 0)]$$

Where K is the number of neighbors we aim to add to each node (depending on the size of the dataset, this is in the order of 100), α (10 by default) and β (3 by default) are expansion parameters that allow more than K nearest neighbors for each node to be initially considered by the balancing process. A weighted directed graph G is constructed using $[a_{ij}]$ as the weighted adjacency matrix. The number of outgoing edges for each node in G is limited by K and the number of incoming edges is bounded by βK. Nodes with lower degrees are however still possible, since outlier

cells may become disconnected or poorly connected during the balancing operations.

Seeding and optimizing a graph cover. We next wish to cover all cells with disjoint and dense subgraphs (or *metacells*) on sizes that are on a scale of the user-defined parameter K used for building the balanced k-NN graph. The K parameter reflects our downstream analysis goals – it should allow for a sufficient accuracy for estimating the UMI distribution within each metacell, but still provide sufficient flexibility to capture multiple sources of variation in the data. Metacells differ conceptually from clusters, since there is no attempt to ensure strong separation between them but only to maximize their coherence. Our algorithm, which is an adaptation of kmeans++ to graphs, starts with a seeding phase that derive an initial set of metacells $I_1 \subset I, .., I_m \subset I$. We denote by $N(i)$ the set of graphic outgoing neighbors of i, and start by defining an empty assignment of cells to metacells $mc(i) = -1$. During seeding iterations, we define the set of covered nodes as $C = \{i \mid mc(i) > -1\}$ and the cover-free score for each node is defined as $f(i) = |N(i) - C|$. We sample seeds as follows:

While $\max_i f(i)/K > size\_min$ do:

sample a new seed j by drawing a sample from cells in $I - C$ with weights $f(i)^3$

update $mc(u) = j \; for \; u \in N(j) - C$

When we meet the stop criterion, cells that are not associated with a seed metacell (i.e. cells for which $mc(i) = -1$) have at most $K * size\_min$ uncovered neighbors, and in particular will almost always has at least one covered neighbor (degree in the balanced graph is typically K, although recall it can be smaller).

Next, define the metacell groups $M_k = \{i \mid mc(i) = k\}$. The association between a cell and a metacell subgraph is based on edges to and from the cell, in a potentially non-symmetric fashion. The outgoing weight vector for each cell is defined as $wo_{im} = \sum_{\{j \in N(i) \cap M_k\}} a_{ij}$. We define $N^{in}(i)$ as the set of incoming neighbors for cell i, and the incoming weight vector is set to $wi_{im} = \sum_{\{j \in N^{in}(i) \cap M_k\}} a_{ji}$. We score metacell association by multiplying these two weights and normalizing by module size, setting $w_{im} = wi_{im} \, wo_{im} / |M_k|^2$. We can now re-assign cells to metacells iteratively until convergence:

Until convergence:

Select a cell i

Reassign $mc(i) = argmax_m w_{im}$

Update weights

These iterations can be done very efficiently when omitting unnecessary recomputation of weights, but the heuristic is not guaranteeing convergence into locally optimal metacell assignment (i.e. where all cells are assigned to their maximum weight metacell). We enforce convergence by employing a cooling strategy: We define a cooling profile $cool(c) = 1 + max(0, c - c_{burn}) * k$, where by default $c\_burn = 10$, and $k = 0.05$. We record the total number of metacell change for each cell as $c(i)$, and modify the weights of the currently assigned metacell to $\mathrm{w_{i,mc(i)}} = \left(\mathrm{w_{i,mc(i)}}\right)^{cool(c(i))}$. Using this approach convergence is guaranteed to occur after a limited number of iterations.

After convergence, there are no formal guarantees on the size of the metacells reported by the algorithm. But, empirically, the seeding process and the connectivity of the graph (K outgoing edges) promote a relatively uniform metacell size and prevent convergence toward solutions with very large subgraphs. The algorithm convergence may be problematic when a large subset of cells (i.e. larger than K) are very homogeneous, which may result in unstable exchange of nodes between several modules covering this subset. In such cases, the cooling strategy described above forces convergence, but some of the modules derived will need to be grouped into more robustly defined clusters in downstream analysis.

Metacell summary statistics. A metacell cover $M_1, .. M_k, .. M_m$ can be studied as a set of meta-transcriptional states by pooling UMI counts:

$$u_{gk} = \sum_{\{i \in M_k\}} u_{gi}$$

$$u_k = \sum_g u_{gk}$$

More precisely, given general assumptions on transcriptional states as sampling from log-normal distributions, and in order to reduce the effect of outliers, we summarize metacells using log transform statistics:

$$p_{gk} = exp\left[\frac{1}{|M_k|}\sum_{\{i \in M_k\}} \log((\epsilon + u_{gi})/(\epsilon G + u_i))\right]$$

$$u'_{gk} = p_{gk} * u_k$$

To compare metacell gene expression (**Figure 3**) we define the log fold change enrichment score:

$$lfp_{gk} = log_2(p_{gk}/median_k(p_{gk}))$$

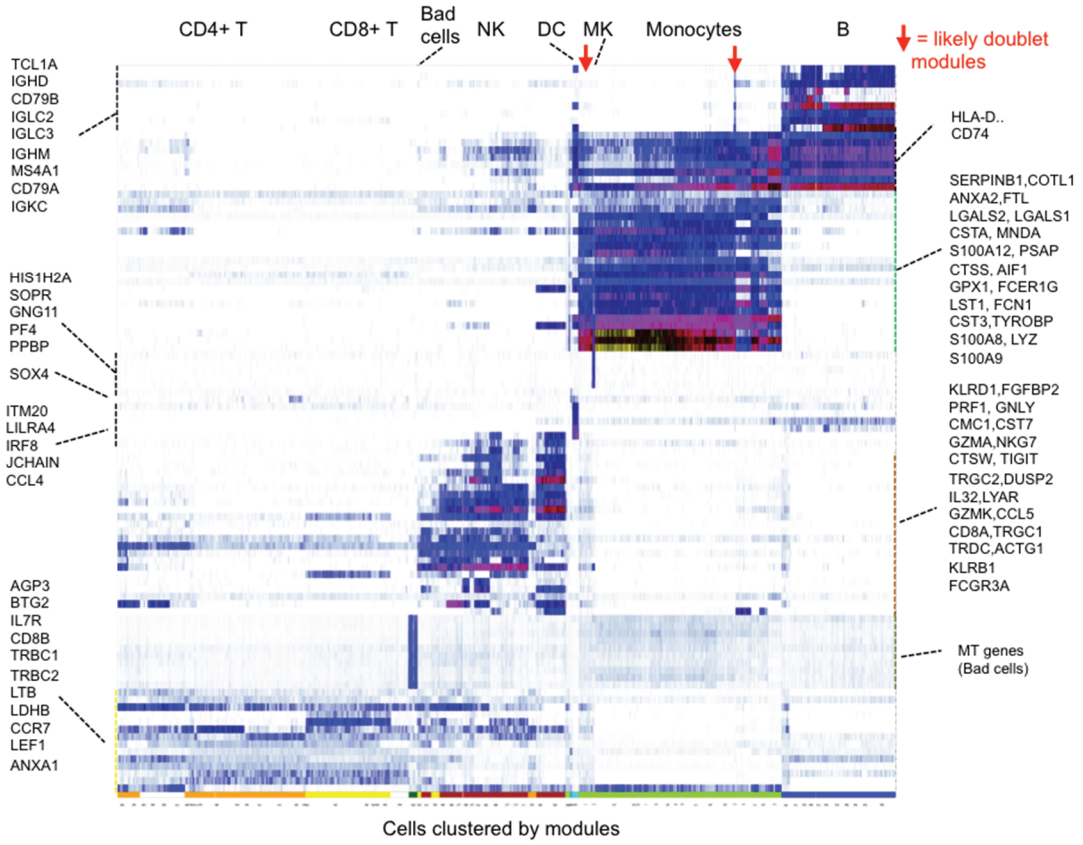

Figure 3. Meta-cells and their marker genes in the PBMC data. Shown are cells (columns), organized into meta-cells (groups of columns), and a set of marker genes (five genes with top fold-change in each meta-cell) with their expression footprint across cells. Color-coding of the bars at the bottom is based on selected marker genes that are known to be associated with specific cell types (Blue – B-cells, Green – Monocyte, Brown – NK cells, Orange/Yellow- T cells, Cyan – DCs, Dark green – Sox4 cells). Cells that are part of a meta-cell with a mixed expression profile suggestive of doublets are marked by red arrows. A cluster of cells/barcodes with high mitochondrial gene expression (and low depths) is marked as "bad cells". See more details on the meta-cell analysis of the PBMC dataset below.

Removing background noise using metacells. The UMI distribution represented by the matrix U is known to be affected by several sources of noise. A background or ambient noise model can be written as:

$$u_{gi} = (1 - \epsilon_g)u_{gi}^{real} + \epsilon_g(\frac{u_g^b}{N^b})$$

where b is the batch of cell i, u_g^b is the total number of molecules observed for the gene in the batch, u_{gi}^{real} is number of molecules of gene g that were actually sampled from cell i, and ϵ_g is a gene-specific noise parameter. This model assumes that the molecules within each cell switch with probability ϵ_g their cellular identity uniformly,

but only within their batch. Such process may represent amplification and sequencing errors of the single cell libraries where barcoded oligonucleotides are priming PCR reaction for molecules that were initially labeled by other cell barcodes ("PCR recombination"). Different genes have different levels of susceptibility to such errors, as determined by their sequence characteristics. Alternatively, this process can account for a homogeneous background concentration of RNA molecules. These molecules are released from cells into the cell suspension during early processing steps and are subsequently labeled and captured as part of the single cell sequencing libraries. Given observations U, it is difficult to infer the noise parameters ε and to distinguish noise from original molecules. Inference of noise is especially difficult when the distribution of gene expression for a gene g is affected by high biological variance, since in this case the distribution u_{gi}^{real} is determined by many parameters and the difference between some inferred u_{gi}^{real} and the empirical UMI distribution becomes under-determined. The noise signature becomes clearer when leveraging metacell assignments, as aggregating cells across metacells stabilizes our estimates of the observed and expected-under-noise expression per gene and per cell. In this way the information collected across multiple genes and cells and stored in the metacell structure can help us estimate in what fraction of cells a given gene is spuriously expressed. To implement this idea we first compute the mean per-cell UMI count for each gene in each batch as:

$$e_{gb} = \frac{u_g^b}{n^b}$$

where u_g^b is the total UMIs for gene g in batch b, and n^b is the number of cells in batch b. Our estimate for the expected background count of each gene in each metacell is then:

$$e_{gm} = \epsilon \sum_b e_{gb} (\sum_i b_{bi} m_{im})$$

where ϵ is some initial guess on the noise level, $B = [b_{bi}]$ is the batch association matrix, set to 1 if cell i is in batch b, and $M = [m_{im}]$ is the metacell association matrix, defined similarly. The observed number of UMIs in a metacell is:

$$o_{gm} = \sum_i u_{gi} m_{im}$$

and the deviation of the expression from the background expectation can be quantified as (using element-wise arithmetic here):

$$z_{gm} = (o_{gm} - e_{gm}) / \sqrt{e_{gm}}$$

We note that in practice we modify the raw U matrix to control the effect of outliers, so that u_{gi}^{reg} is used instead of u_{gi} when computing o_{gm}:

$$u_{gi}^{reg} = min(u_{ij}, \lceil e_{gb} \rceil + 3\sqrt{\lceil e_{gb} \rceil})$$

Rounding up is performed in order to consider low UMI count genes conservatively.

We next introduce our main assumption regarding background/ambient expression. A gene is considered spuriously expressed if it is expressed in a level no higher than the noise prediction in a sufficiently large set of cells. Note that a uniformly expressed (i.e. housekeeping) gene is not expected to be expressed at $\epsilon * U$ levels in any metacell, unless the expected expression is very small and sampling variance is dominating the signal even after pooling in metacells is performed. Specifically, we define for each gene the set of metacells that are compatible with the background model using the Boolean matrix:

$back_{gm} = z_{gm} < T$ (T=2 by default)

Then we test, for each gene, if the total number of cells with $back_{gm} = TRUE$ is at least α of the total number of cells, and if the total number of UMIs for the gene in such cells is at most $2 * \epsilon * u_g$. If this is the case, we mark the gene as a valid candidate for filtering ambient noise, generating a set G_{tofilt}. To perform UMI filtering for such genes in practice, we set $u_{gi} = 0$ for g in G_{tofilt} and cells contained in metacells with $back_{gm} = 1$. We are however not filtering outlier genes (those with $u_{gi}^{reg} < u_{gi}$).

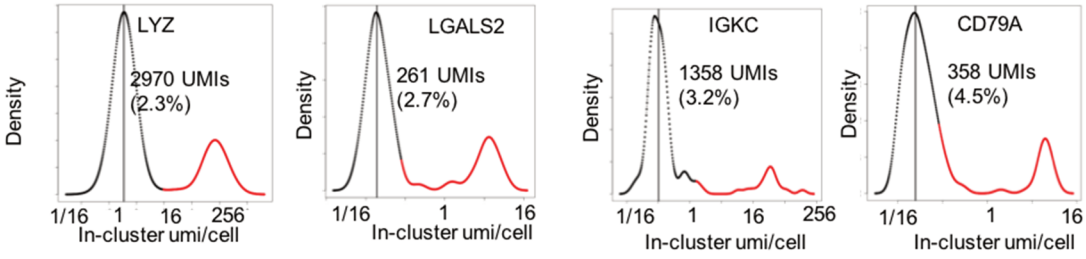

Figure 4. Filtering ambient UMIs. Shown are distributions of mean expression per cell over meta-cells for four highly expressed and heterogeneous genes in the PBMC dataset. Using the uniform ambient noise model introduced here, we identify clusters that show expression of these marker genes at levels consistent with a constant small fraction of the total UMIs being distributed uniformly (black colored part of the distribution). The marker UMIs in them are removed. This policy leads to simple and reliable filtering for very strong markers (e.g. LYZ, IGKC, with a mean expression on positive meta-cells > 20). Weaker genes (e.g. LGALS2 and CD79A) may be more sensitive to such filtering, and in these cases a conservative threshold is being used. We note that ambient contamination is of particular impact on downstream analysis when analyzing together several batches, each with a different mixture of cell states/types.

The above procedure is sensitive to the initial parameters used. We use a minimal background subpopulation fraction α =1/8. In application to MARS-seq analysis, an initial guess of epsilon as ϵ = 0.03 is appropriate. For the PBMC 10x data illustrated below, a higher level of noise is observed and we used ϵ = 0.06 (**Figure 4**). We note that setting a higher initial epsilon is not affecting accuracy of filtering for high expression noise, but can result in too aggressive filtering in low expression noise. In general, cleaning ambient/background UMIs should be approached carefully to ensure no additional biases are introduced into the matrix. However, in cases of a multi-batch dataset, in which the cell population per batch is variable and contribute to variability in the background noise distribution, filtering noise using metacells can reduce batch bias without filtering key genes.

Identifying batch-affected genes. Technical batch effects in scRNA data may be difficult to characterize when studying multiple samples coming from different sources. This is because the cell-type or cell-state composition in each batch may be different, representing a biological sample-specific variation that can combine with technical biases in ways that are difficult to decouple at the single gene level. However, once cells are grouped into metacells we can increase the robustness of batch effect estimation by aggregating information across cells within metacells, analogously to our strategy for removing background noise. Again $B = [b_{ib}]$ is the batch association matrix, $M = [m_{im}]$ is a module association matrix and we compute the fraction of each gene in each metacell as:

$$f_{gm} = \sum_i u_{gi} m_{im} / u_m$$

where u_m is the number of UMIs per metacell. The total number of UMIs per batch per metacell is:

$$u_{mb} = \sum_i u_i m_{mi} b_{ib}$$

Assuming a null model in which genes are expressed at metacell-specific levels with no batch effects, our estimate for the expected number of UMIs per gene per batch is:

$$e_{gb} = \sum_m f_{gm} u_{mb}$$

and the observed number is:

$$o_{gb} = \sum_i u_{gi} b_{ib}$$

giving a ratio of:

$br_{gb} = (o_{gb} + u_{reg})/(e_{gb} + u_{reg})$

where u_{reg} is a regularization constant, set to 10 UMIs by default.

Genes with high levels of $|log2(br_{gb})|$ may be considered for blacklisting during feature selection as discussed above. Consideration of batch bias is otherwise done as part of downstream analysis, when testing specific hypotheses on gene regulation within and between metacells. We generally avoid direct normalization of the sparse UMI matrix given the br ratios, to avoid introduction of systematic bias (normalizing by constant factors) or additional sampling noise in the U matrix (if sampling the batch effects out).

Resampling for detection of robust clusters and filtering poorly clustered cells. The greedy graph cover algorithm we outlined above is designed to dissect complex single cell populations efficiently, but it provides no guarantees on the robustness of the derived metacells. In particular, when aiming to detect strongly separated *clusters* of cells in the data, the grah cover provide us with only building blocks from which clusters can be built. To assess metacell robustness we use a bootstrap approach, in which we repeatedly compute graph covers for random subsets (e.g. 75%) of the cells. We summarize the resampled metacells for each cell subset sample in matrices $O = [o_{ij}]$ and $C = [c_{ij}]$ that specify how many times the pair of cells i, j were sampled and how many times they were both assigned to the same metacell, respectively. Given any set of metacells $M = [m_{im}]$, we can assess the consistency of metacell association for a cell i using the probability of co-clustering i with m:

$c_{im} = \sum_j \frac{c_{ij} m_{jm}}{c_i}$

here c_i is the total number of co-cluster observations for cell i. Given a robust set of metacells we expected c_{im} values to be close to 1, and we can filter cells with weak association to their cluster by setting a minimal threshold on c_{im}. We note that the graph cover algorithm discussed above will split large strong clusters into several metacells, decreasing probability of co-clustering for each individual metacell even when the set of metacells is highly robust as a group.

We can also rely on the resampling approach for identifying well-separated cell clusters. We generate the co-clustering similarity matrix:

$S^{boot} = C / O$ (element-wise division)

and use the Pearson correlations between columns in S^{boot} to re-cluster cells in a simple hierarchical clustering scheme, or through other standard clustering approaches (e.g. louvain modularity). When this approach is used, the final clustering resolution depends on user decision since several levels of robust clustering may be present in the data. User choice will greatly depend on the biological goals of the analysis, since the maximal resolution that is robust given the data is not always the resolution most appropriate for further characterization. **Figure 5** shows the bootstrap co-clustering matrix derived for the PBMC dataset, which we used in order to extract 80 robust clusters.

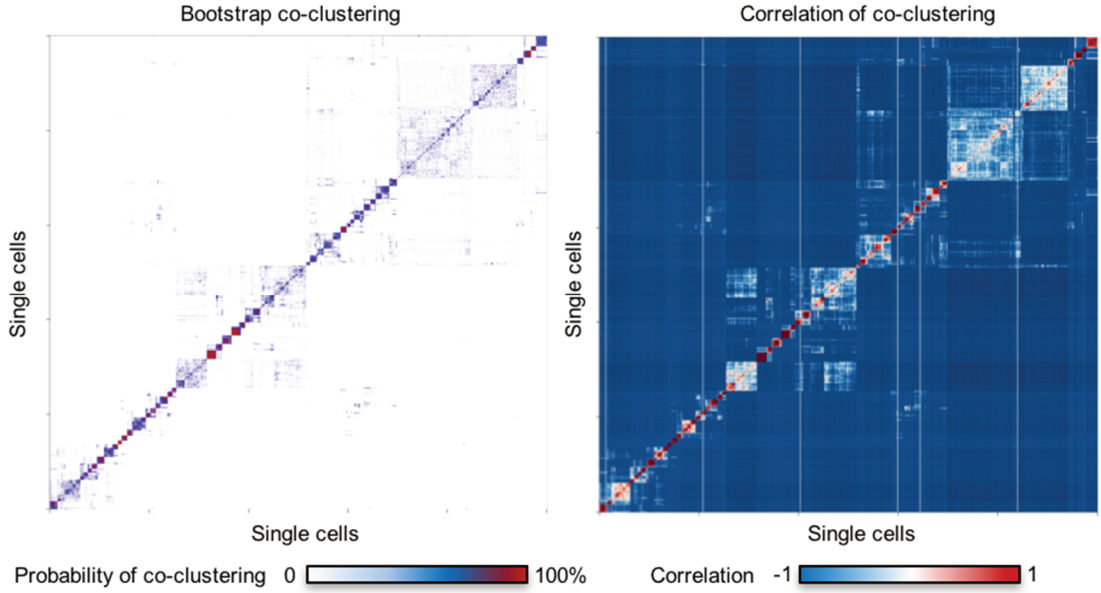

Figure 5. Assessing meta-cell robustness using resampling. Shown are results from 500 iterations of resampling 75% of the PBCM data and recomputing graph cover solutions. The matrix on the left summarizes for each pair of cells the probability of co-occurrence in a meta-cell. The matrix on the right shows the correlation between the co-occurrence vectors of each cell, which can be used to re-cluster cells hierarchically and derive cell clusters. In this case, 80 clusters were identified in the data and used for visualization and downstream analysis below.

The non-parametric robustness analysis that we described above can be supplemented with a heuristic for filtering cells that are incompatible with their metacell's gene expression signature. This is done by first identifying for each metacell the set of enriched genes ($G_m = g$ s.t. $fp_{gm} > 2$), and then computing the distribution of total number of UMIs of genes G_m in each cell. We divide the G_m expression by a the median over all cells, and assign each cell in the metacell with an enrichment ratio. Small enrichment ratios are indicative of poor linkage between the cell and the metacell transcriptional signature, and can be used for filtering cells even when their a-parametric co-clustering score is high. For the PBMC data illustrated here we used the hierarchical clustering of the co-clustering bootstrap matrix S^{boot} to define 80 clusters. We used these clusters for visualization below.

Filtering doublets and outlier cells. A bootstrap scheme for testing metacell robustness does not control for cases where a cell is robustly linked by k-nn similarities to a metacell, but also expressed significantly genes that are not observed in the other neighboring cells and is therefore violating the rationale underlying the metacell concept, aiming at groups of transcriptionally homogeneous cells. Such a scenario can be indicative of rare cell types or subtypes that combine with more frequent cell states to form an inappropriately heterogeneous metacell. It can also represent cell doublets that become associated with a single barcode due to various technical errors. We can search for cellular outliers that do not adhere to their metacell's expression distribution by first estimating the multinomial UMI distribution in each metacell:

$$p_{gm} = \sum_i u_{gi} m_{im} / u_m$$

here u_m is the total number of molecules in all cells of the metacell, and normalization is done per column. We then estimate the expected number of molecules per cell and gene, under the null hypotheses of equal expression in all cells per metacell:

$$e_{gi} = u_i \sum_m p_{gm} m_{im}$$

and test for outliers using either a regularized Z-score:

$z_{gi} = (u_{gi} - e_{gi}) / \sqrt{e_{gi} + reg}$ (e.g., reg=1)

or fold change

$out_{gi} = (u_{gi} + 1) / (e_{gi} + 1)$

The maximum Z score or fold change for each cell can be used to detect significant outliers. **Figure 6** illustrates outlier detection in the PBMC dataset, using a threshold of $out_{gi} > 10$, highlighting several cases of likely doublet barcodes merging megakaryocytes/platelets with different cell types, rare immunoglobulin patterns and several additional outliers. We note that a threshold of 10 fold is used to ensure we detect only extreme cases of genes that are highly over-expressed over the expected metacell's expression level, and that we do not aim to enforce a strictly multinomial distribution within each metacell in a dataset of only 8,000 cells. As the number of cells increase, more rigorous (e.g. lower fold change threshold) testing for outliers would become desirable

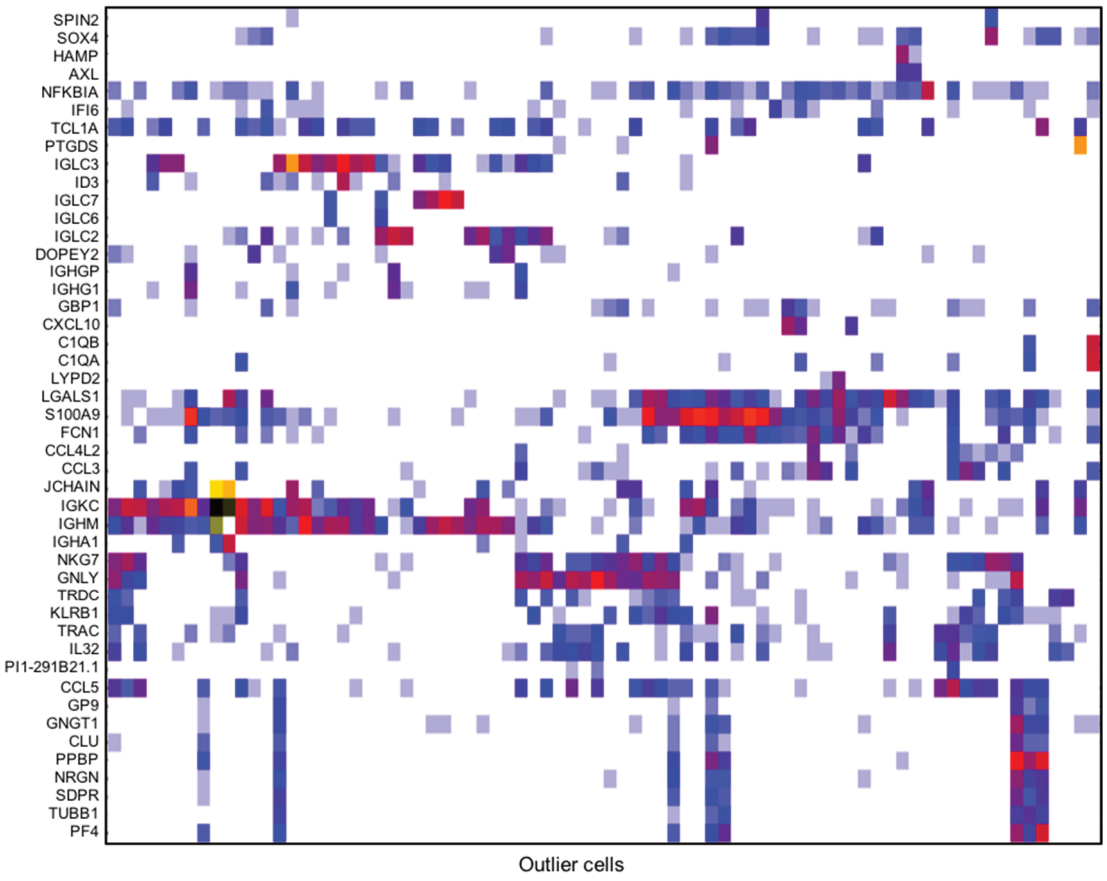

Figure 6. Detecting outlier cells. Shown are cells detected as meta-cell outliers (columns) and the genes that defined their statistical divergence from the predicted UMI distributions of their meta-cell (rows). Cells and genes are clustered hierarchically, but as expected, the emerging structure is of multiple combinations of unrelated marker genes, reminiscent of the distribution expected from rare doublets. In this case, megakaryocyte-associated genes (PF4, PPBP) are frequently co-expressed with markers specific to diverse other cell types (e.g. IGKC, S100A9), suggesting doublets are affected by specific cell type tendency to form physical pairings. We note that combinations of cell types that are more frequent in the data are likely to form their own specific meta-cells, and these "doublet" meta-cells must be detected separately and not as individual cells (specific doublet meta-cells that were detected manually are highlighted in Figure 3).

Metacell regularized force directed 2D projection. Large scRNA-seq datasets can give rise to complex metacell covers that must be further interrogated for higher order structures. Projection of metacells (and the cells belonging to them) in 2D space can provide one avenue for exploring the similarities between cells and metacells. To derive a metacell 2D projection, we use the balanced similarity graph G and summarize the total number of (unweighted) edges linking metacells :

$$B = b_{ml} = \frac{K^2}{|M_m| * |M_l|} \sum_{\{i \in M_m, j \in M_l\}} \lceil a_{ij}/K \rceil$$

$K = median(n_m)$ is a scaling constant. See **Figure 7** for an illustration of such matrix.

We normalize B rows and columns:

$$b_m = \sum_l b_{ml}\, b_{ml}^{out} = \frac{b_{ml}}{b_m},\ b_l = \sum_m b_{ml}\ ,\ b_{ml}^{in} = \frac{b_{ml}}{b_l}$$

And define the score for connecting two metacells as $b'_{ml} = b_{ml}^{in} + b_{ml}^{out}$. We retain as candidate edges only pairs for which $b'\_ml > T\_edge$. We then construct a graph $G^M = (M, E^M)$ on metacells M=(1,..,m), by adding the D highest scoring candidate edges (if such edges exists) for each metacell. Note that any metacell in the graph can be completely disconnected if its cells are highly connected to themselves but not to any other metacell. We project the metacell graph into 2D using a standard force-directed layout algorithm, derive coordinates for each metacell (x_k, y_k). We also position each cell i using average position of the metacells of neighboring cells :

$$x_i = \frac{1}{Z} \sum_{\{j \mid a_{ij} > W^{layout}, (mc(i), mc(j) \in E^M\}} x_{\{mc(j)\}}, y_i = \frac{1}{Z} \sum_{\{j \mid a_{ij} > W^{laout}\ (mc(i), mc(j) \in E^M\}} x_{\{mc(j)\}}$$

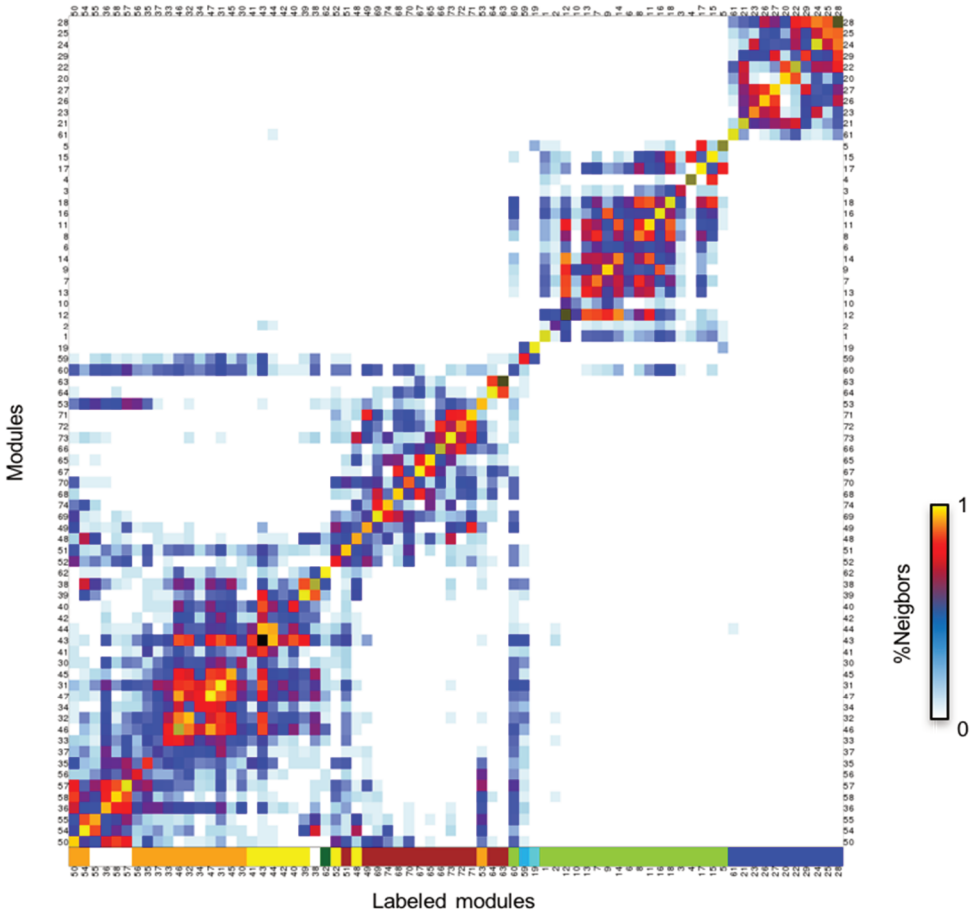

Figure 7. Connecting meta-cells by common K-nn neighbors. The matrix is indicating the fraction of K-nn cell-cell adjacencies that bridge any pair of meta-cells in the PBMC dataset. The diagonal represent the robustness of the meta-cell on the K-nn graph. The labels on the bottom are based on identification of a specific list of markers associated with the major cell types in the data.

Where W^{layout} is a parameter determining how many edges we will use to position a cell (0 will imply using all edges). In practice we add some Gaussian noise to both coordinates to minimize cell overlap. Note that a metacell with perfect clustering will imply all of its cells will be positioned precisely on one specific center.

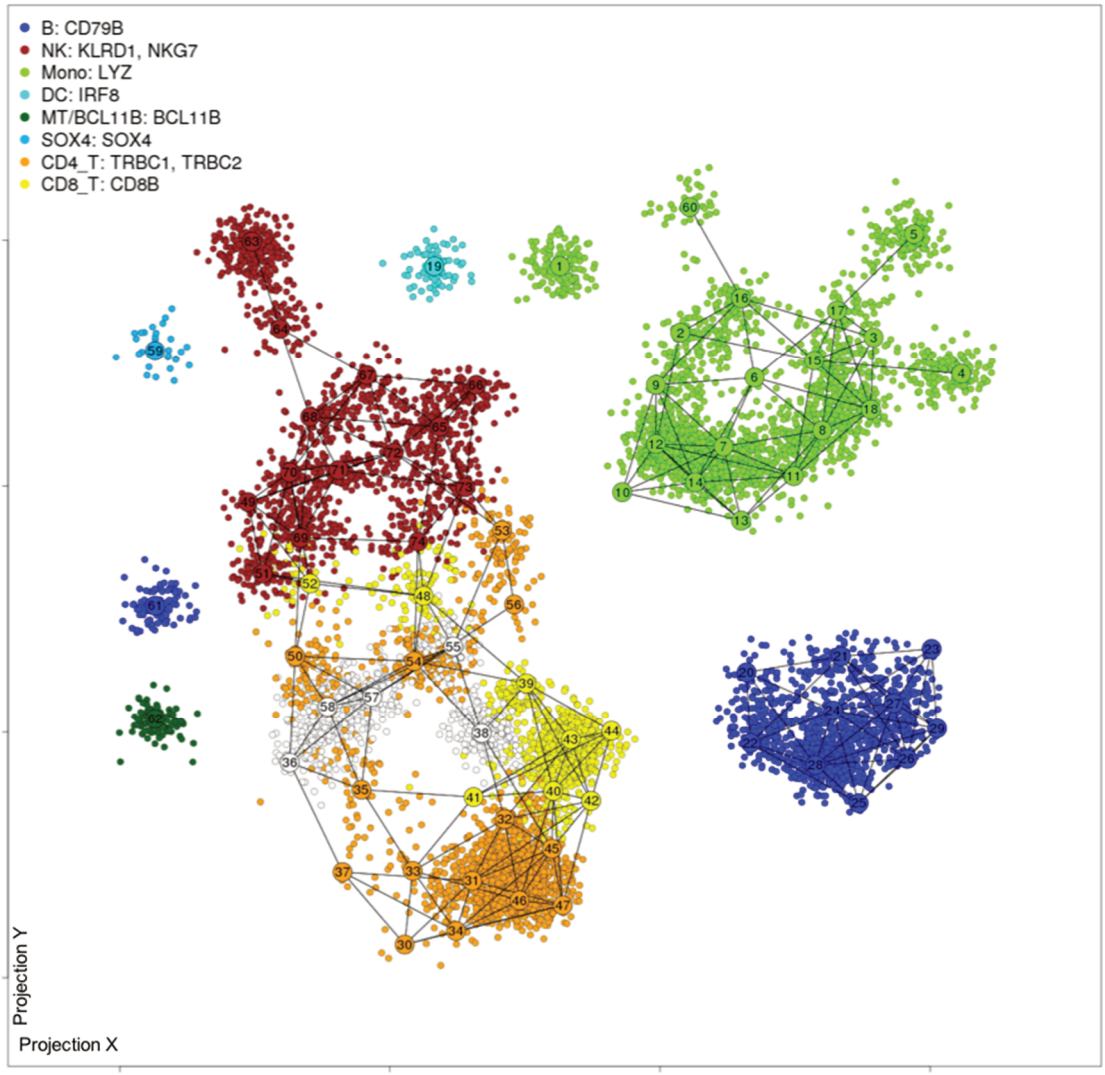

Figure 8. Meta-cell graph-based 2D layout. The meta-cell graph is visualized using numbered color coded circles and edges that connect them. Cells are depicted as small color coded dots around meta-cells, positioned according to the coordinates of the meta-cells associated with their K-nn cells. Color coding is done using a list of marker genes, where each meta-cell is associated with the color of the marker showing the highest enrichment in its cells. The color codes correspond to those used in Figure 7.

Projecting metacells, cells and genes in 2D. We project the metacell graph into 2D using a standard force-directed algorithm. We note that the graph is constructed such that the dependency structure is simplified and the projection is more likely to remain coherent. Given the projected metacell graph coordinates and the original cell graph G, we position each cell i relative to its metacell mod_i coordinates. This is done by counting the number of edges linking i with cells in each of the metacells that are

connected to mod_i in the metacell graph (including mod_i itself), computing the a weighted average respective graph nodes coordinates, adding Gaussian noise to avoid overlaps. **Figure 8** shows the projected module cover for the PBMC dataset.

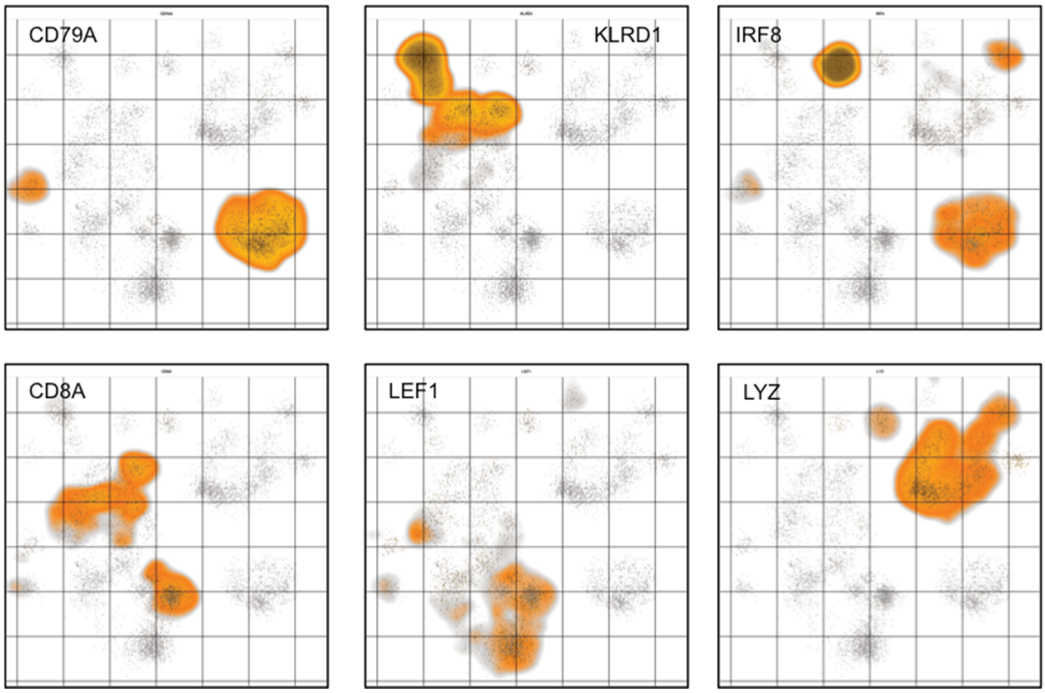

Figure 9. Projecting genes on 2D meta-cell map. Examples visualizing the distribution of six marker genes on the PBMC projected 2D map.

To plot the distribution of genes in the projected 2D space, we position each molecule in u_{gi} according to the computed i coordinate and use a kernel density function to derive densities for 2D bins given all molecule coordinates. We normalize this density by the density of all cells in the metacell (assuming each cell contributes one molecule) (**Figure 9**).

DISCUSSION

We described an integrated approach for analyzing single cell RNA-seq data, which addresses the sparse characteristics of single cell profiles through the identification of metacells – groups of cells with coherent expression distributions. Metacells can be regarded as a non-parametric piecewise approximation of the complex gene expression space, or as a way to compute *meta-states* that represent single cells quantitatively at higher resolution than possible using the raw data. Since each cell contributes to at most one metacell the statistics of molecule distributions between and within metacells are easily interpretable. The loss of accuracy due to grouping of cells with potential variable expression has been decreasing as more cells are being analyzed in a typical study, and through accumulating cells from multiple studies.

When viewed as an approximation for the expression distribution, a metacell cover facilitates robust procedures for identifying background noise, for finding outlier cells and batch effects, and is a useful component in a boostrap approach for finding clusters. The approach also allows visualization of the data by summarizing the most important similarity relationships into a graph structure that can be projected in 2D.

Several existing approaches for scRNA-seq analysis implement elements of the pipeline described here. In particular, several algorithms model single cell cohorts using a K-nn similarity graph and transform the resulting graph structure in order to project cells in 2D and to infer possible dynamical processes from stationary observations. The metric used to constructed the graph vary between tools, but in most cases there is a preliminary stage of dimensionality reduction (e.g. using PCA). In some of these algorithms, the model is applied to individual cells, resulting in a massive statistical object (the full K-nn graph) that is difficult for application of simple statistical tests on model fitting or noise as described here for Metacells. Standard clustering approaches are being applied to scRNA data as well, but in many cases, clustering the raw data directly is affected by multiple (and sometime contrasting) biological effects (differentiation, stress, cell cycle, batch effects) and this may lead to ambiguous outcomes. Our approach avoids assumptions on the parametric relationship between metacells, and utilizes metacells as a generic statistical device rather than as a component in a modeling scheme. This provides us with a powerful tool for taking the multivariate distribution of gene expression into account when addressing low-level questions on noise, duplications, outliers and bias. We believe that a multivariate approach for these questions is significantly more powerful than algorithms using parametric assumptions on individual genes. Approaches for detecting block structures[23] and micro-clustering[24] provide additional avenues for further developing bottom up algorithms for modelling complex single cell RNA-seq datasets, employing further hierarchical and/or parameteric assumptions that are not used in the present work.

We hypothesize that in the coming years, exciting questions and modeling strategies that aim at inferring developmental dynamics, comparing samples, building spatial maps and much more will be ideally approached by replacing raw single cell data by meta-states that combine data from multiple (hundreds) of single cell observations. Such meta-states would ideally be supported by multiple replicated experiments and can be organized systematically and exchanged between studies and groups. Robustly inferred meta-states can allow us to separate low-level questions on data

processing and noise filtering from biological questions on temporal and regulatory processes. While the graph cover strategy we introduce here can serve as a starting point in developing such strategies for meta-state inference, it will be important to expand and improve the algorithms generating such covers, such that their performance will scale favorably as the single cell genomics community continues to chart unknown territories of cellular transcriptional programs at single cell resolution.

LITERATURE CITED

1. Buettner, F. *et al.* Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells. *Nat. Biotechnol.* **33,** 155–160 (2015).
2. Grün, D. *et al.* Single-cell messenger RNA sequencing reveals rare intestinal cell types. *Nature* **525,** 251–255 (2015).
3. Jaitin, D. A. *et al.* Massively parallel single-cell RNA-seq for marker-free decomposition of tissues into cell types. *Science* **343,** 776–9 (2014).
4. Kumar, R. M. *et al.* Deconstructing transcriptional heterogeneity in pluripotent stem cells. *Nature* **516,** 56–61 (2014).
5. Levin, M. *et al.* The mid-developmental transition and the evolution of animal body plans. *Nature* **531,** 637–641 (2016).
6. Macosko, E. Z. *et al.* Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets. *Cell* **161,** 1202–1214 (2015).
7. Reinius, B. *et al.* Analysis of allelic expression patterns in clonal somatic cells by single-cell RNA–seq. *Nat. Genet.* **48,** 1430–1435 (2016).
8. Zeisel, A. *et al.* Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science* **347,** 1138–1142 (2015).
9. Tanay, A. & Regev, A. Scaling single-cell genomics from phenomenology to mechanism. *Nature* **541,** 331–338 (2017).
10. Bendall, S. C. *et al.* Single-Cell Trajectory Detection Uncovers Progression and Regulatory Coordination in Human B Cell Development. *Cell* **157,** 714–725 (2014).
11. Fan, J. *et al.* Characterizing transcriptional heterogeneity through pathway and gene set overdispersion analysis. *Nat. Methods* **13,** 241–244 (2016).
12. Grün, D. *et al.* De Novo Prediction of Stem Cell Identity using Single-Cell Transcriptome Data. *Cell Stem Cell* **19,** 266–277 (2016).
13. Haghverdi, L., Buettner, F. & Theis, F. J. Diffusion maps for high-dimensional single-cell analysis of differentiation data. *Bioinformatics* **31,** 2989–2998 (2015).
14. Ilicic, T. *et al.* Classification of low quality cells from single-cell RNA-seq data. *Genome Biol.* **17,** 29 (2016).
15. Lun, A. T. L., McCarthy, D. J. & Marioni, J. C. A step-by-step workflow for low-level analysis of single-cell RNA-seq data with Bioconductor. *F1000Research* **5,** 2122

(2017).

16. Lun, A. T. L. & Marioni, J. C. Overcoming confounding plate effects in differential expression analyses of single-cell RNA-seq data. *Biostatistics* **18,** 451–464 (2016).
17. Setty, M. *et al.* Wishbone identifies bifurcating developmental trajectories from single-cell data. *Nat. Biotechnol.* **34,** 637–645 (2016).
18. Stegle, O., Teichmann, S. A. & Marioni, J. C. Computational and analytical challenges in single-cell transcriptomics. *Nat. Rev. Genet.* **16,** 133–145 (2015).
19. Svensson, V. *et al.* Power analysis of single-cell RNA-sequencing experiments. *Nat. Methods* **14,** 381–387 (2017).
20. Vallejos, C. A., Risso, D., Scialdone, A., Dudoit, S. & Marioni, J. C. Normalizing single-cell RNA sequencing data: challenges and opportunities. *Nat. Methods* **14,** 565–571 (2017).
21. Satija, R., Farrell, J. a, Gennert, D., Schier, A. F. & Regev, A. Spatial reconstruction of single-cell gene expression data. *Nat. Biotechnol.* **33,** 495–502 (2015).
22. Wagner, A., Regev, A. & Yosef, N. Revealing the vectors of cellular identity with single-cell genomics. *Nat. Biotechnol.* **34,** 1145–1160 (2016).
23. Peixoto, T. P. Hierarchical Block Structures and High-Resolution Model Selection in Large Networks. *Phys. Rev. X* **4,** 11047 (2014).
24. Zheng, S., Papalexi, E., Butler, A., Stephenson, W. & Satija, R. Molecular transitions in early progenitors during human cord blood hematopoiesis. *Mol. Syst. Biol.* **14,** e8041 (2018).