CALISTA: Clustering and Lineage Inference in Single-Cell Transcriptional Analysis

Nan Papili Gao[1,2], Thomas Hartmann[1], Tao Fang[1], Rudiyanto Gunawan[1,3,*]

[1] Institute for Chemical and Bioengineering, ETH Zurich, Zurich, 8093, Switzerland
[2] Swiss Institute of Bioinformatics, Lausanne, 1015, Switzerland
[3] Department of Chemical and Biological Engineering, University at Buffalo, Buffalo, NY, 14260 USA

* To whom correspondence should be addressed. Tel: +1 716 645 0952; Fax: +1 716 645 3822; Email: rgunawan@buffalo.edu

ABSTRACT

We present CALISTA (Clustering and Lineage Inference in Single-Cell Transcriptional Analysis), a numerically efficient and highly scalable toolbox for an end-to-end analysis of single-cell transcriptomic profiles. CALISTA includes four essential single-cell analyses for cell differentiation studies, including single-cell clustering, reconstruction of cell lineage specification, transition gene identification, and cell pseudotime ordering. These analyses can be applied individually or in a pipeline. CALISTA is developed using a likelihood-based approach based on probabilistic models of stochastic gene transcriptional bursts and random technical dropout events. We illustrate the efficacy of CALISTA using single-cell gene expression datasets from different single-cell transcriptional profiling technologies and from a few hundreds to tens of thousands of cells. CALISTA is freely available on https://github.com/CABSEL/CALISTA.

INTRODUCTION

The differentiation of stem cells into multiple cell types relies on the dynamic regulation of gene expression (1). In this regard, advances in single-cell gene transcriptional profiling technology have given a tremendous boost in elucidating the decision making process governing stem cell commitment to different cell fates (2). The applications of single-cell transcriptional analysis have led to new insights on the functional role of cell-to-cell gene expression heterogeneity in the physiological cell differentiation process (3–7). Along with the surge in single-cell transcriptional profiling studies, algorithms for analyzing single-cell transcriptomics data have received increasing attention. In comparison to measurements from aggregate or bulk samples of cell population, single-cell gene expression profiles display much higher variability, not only due to technical reasons, but also because of the intrinsic stochastic (bursty) dynamics of the gene transcriptional process (8). In particular, the

stochastic gene transcription has been shown to generate highly non-Gaussian mRNA count distributions (9), which complicate data analysis using established material and methods that rely on a standard noise distribution model (e.g. Gaussian or Student's *t*-distribution).

Numerous algorithms have recently been developed specifically for the analysis of single-cell gene expression data. A class of these computational algorithms is geared toward identifying cell groups or clusters within a heterogeneous cell population. Traditional clustering algorithms such as *k*-means and hierarchical clustering have been applied for such a purpose (10–12). Several other single-cell clustering strategies, such as CIDR (13), pcaREDUCE (14), scLVM (15) and SNN-cliq (16), adapt more advanced algorithms such as nearest neighbors search. Time-variant clustering strategies have also been implemented to elucidate the appearance of multiple cell lineages (17, 18). In addition, consensus clustering material and methods, such as SC3 (19) and SIMLR (20), have received much interest thanks to their superior stability and robustness. Finally, a likelihood-based method called SABEC (Simulated Annealing for Bursty Expression Clustering) (21) employs a mechanistic model of the bursty stochastic dynamics of gene transcriptional process to cluster cells.

Another important class of algorithms deals with the reconstruction of lineage progression during cell differentiation process and the pseudotemporal ordering of single cells along the cell developmental path(s). The lineage progression describes the transition of stem cells through one or several developmental stages during the cell differentiation. This progression may comprise a single developmental path from the progenitor cells to one final cell fate, as well as bifurcating paths leading to multiple cell fates. In this class of algorithms, the reconstruction of the lineage progression and developmental paths is commonly implemented for the purpose of pseudotemporal cell ordering. The pseudotime of a cell represents the relative position of the cell along the developmental path and is typically normalized to be between 0 and 1. By plotting the gene expression against the pseudotime of the cells along a developmental path, one obtains a dynamic trajectory of the gene expression based on which the gene regulations driving the cell fate decision-making can be inferred. Numerous algorithms are available for single-cell transcriptional data analysis for cell lineage inference and cell ordering, notably DPT (22), MONOCLE (23, 24) and SCUBA (17) (for a more complete list, see a recent review by Canoodt et al. (25)).

In this work, we developed CALISTA (Clustering and Lineage Inference in Single Cell Transcriptional Analysis), a numerically efficient and highly scalable toolbox for an end-to-end analysis of single-cell transcriptomics data. CALISTA is capable of and has been tested for analyzing datasets from major single-cell transcriptional profiling technologies, including RT-qPCR, single-cell RNA-sequencing (scRNA-seq) and single-cell Drop-sequencing (scDrop-seq). CALISTA enables four essential analyses of single-cell transcriptomics in stem cell differentiation studies, namely single-cell clustering, reconstruction of lineage progression, transition gene identification and cell pseudotime ordering. In existing literature, these analyses are typically carried out by stringing several task-specific tools together in a bioinformatics pipeline. But, the basic assumptions behind different tools (e.g., regarding the

distribution of data noise) maybe incompatible, an issue that has not been delved into more carefully in the literature. In contrast, the different analyses in CALISTA are compatible with each other as they are based on the same likelihood-based approach using probabilistic models of gene transcriptional bursts and random dropout events.

In the next section, we describe the algorithmic aspects and functionalities of CALISTA. Subsequently, we illustrate CALISTA's end-to-end analysis using single-cell transcriptional profiles from the differentiation of human induced pluripotent stem cells (iPSCs) into mesodermal (M) or undesired endodermal (En) cells (26). Further, we compare the performance of CALISTA to that of popular bioinformatics tools for single-cell clustering and cell pseudotime ordering, including SIMLR (20), SC3 (19), DPT (22), SCUBA (17), and MONOCLE2 (23, 24), in analyzing a diverse set of previously published single-cell transcriptional profiles. Finally, we demonstrate the scalability of CALISTA in analyzing large datasets from scDrop-seq studies.

MATERIAL AND METHODS

Input data and data preprocessing

The single-cell gene expression matrix should be formatted into an $N \times G$ matrix, where *G* denotes the number of genes, *N* denotes the number of cells, and the matrix element $m_{n,g}$ is the transcriptional expression value of gene *g* in the *n*-th cell. CALISTA accepts expression values from RT-qPCR (2^{C_t} value), scRNA-Seq (e.g. log(RPKM) or log(TPM)) or scDrop-Seq (e.g. gene UMI counts). In preparing the input matrix, the single-cell gene expression data should first be normalized (e.g. by spike-ins) (3, 27), transformed into the appropriate expression values (28, 29) and removed of batch effects (30). For Drop-Seq, the UMI data are further scaled by dividing each gene UMI count with the total UMI count in the corresponding cell and then multiplying the value with the median of the total UMI counts among cells (31).

Before performing single-cell analysis, we first preprocess the single-cell expression matrix by removing the genes and cells (i.e., columns and rows of the expression matrix, respectively) with a large fraction of zero expression values, exceeding a user-defined threshold (default threshold: 100% for genes and 100% for cells) (Figure 1a). For scRNA-seq datasets, CALISTA further selects a number of informative genes *Y* for the single-cell analysis following a previously described procedure (32). Briefly, CALISTA first calculates gene-wise mean (average) expressions and coefficients of variation (CV) across the cells. The genes are partitioned into 20 bins based on their mean expressions. The CV values of the genes within each bin are subsequently transformed into z-scores. Finally, the genes from the *Y* largest CV z-scores among all bins are selected for the single-cell analysis. The number of genes *Y* is set to the minimum among the following: 200, the number of genes *G*, and a user-defined percentage of the number of cells *N*.

For scDrop-Seq datasets, genes are again binning genes into 20 bins based on their mean expression. We first select the top *Y* genes in decreasing magnitude of the normalized

dispersion of the genes by following the procedure described for droplet-based data in Seurat (32).

Next, CALISTA scales the single-cell expression values such that the maximum value of any gene is 200. The scaling is carried out as follows:

$$\hat{m}_{n,g} = 2^{\frac{log_2(m_{n,g}+1)}{log_2(m_{max,g})} log_2(200)} \qquad (1)$$

where $m_{n,g}$ and $\hat{m}_{n,g}$ represent the original and scaled expression value of gene *g* in the *n*-th cell respectively, and $m_{max,g}$ is the maximum expression value in gene *g* (i.e. the maximum of $m_{n,g}$ over all cells). The scaling above allows CALISTA to use a pre-computed table for the maximum likelihood in the clustering analysis, thereby reducing the computational cost significantly (see **Cell Clustering** below). We tested using *in silico* datasets and confirmed that the scaling of the expression values does not affect the clustering accuracy (see Supplementary Note S1, Figure S1 and Table S1). After scaling, for scDrop-Seq data, we re-rank the top *Y* genes in increasing value of the gene-wise likelihood v^g computed following Equation (13) below, with the random dropout model in Equation (4) incorporated in the calculation of likelihood. A lower gene-wise likelihood value indicates a broader distribution of single-cell expression. Genes with likelihood values exceeding a given threshold (by default set at the elbow of the curve of likelihood vs gene rank) is removed from further analysis.

Cell Clustering

As illustrated in Figure 1b, CALISTA combines maximum likelihood and consensus clustering algorithms for single-cell clustering analysis. Following a previous method SABEC (21), CALISTA uses the stochastic two-state model of gene transcription to derive the steady-state probability distribution of gene expression values. The probability distribution is used to prescribe a likelihood value to a cell based on the measured single-cell expression values of its genes (21). In contrast to SABEC that employs a simulated annealing algorithm, CALISTA implements a numerically efficient greedy algorithm to solve the likelihood maximization problem. By repeatedly running the greedy algorithm, each starting from a different random initial cell assignment, CALISTA generates a consensus matrix containing the number of times that any two cells are assigned to a cluster. Finally, CALISTA uses the consensus matrix to cluster cells by *k*-medoids (33).

Stochastic two-state gene transcriptional model

The two-state model of stochastic gene transcriptional process in CALISTA follows the model developed by Peccoud and Ycart (34). The model describes a promoter that switches stochastically between an OFF (inactive or non-permissive) state, where the gene transcription cannot start, and an ON (active or permissive) state, where the gene

transcription can proceed. The set of reactions describing the stochastic gene transcription in the two-state model are as follows:

$$OFF \xrightarrow{\theta_{on}} ON$$

$$ON \xrightarrow{\theta_{off}} OFF$$

$$ON \xrightarrow{\theta_t} ON + mRNA$$

$$mRNA \xrightarrow{\theta_d m} \emptyset$$

where θ_{on} is the rate of the promoter activation, θ_{off} is the rate of the promoter inactivation, θ_t is the rate of mRNA production when the promoter is active, θ_d is the rate constant of mRNA degradation, and *m* denotes the number of mRNA molecules. At steady state, the probability distribution of mRNA count *m* can be approximated by the following density function (9):

$$P_2(m;\ \theta_{on}, \theta_{off}, \theta_t) = \frac{\Gamma(\theta_{on} + m)\Gamma(\theta_{on} + \theta_{off})\theta_t^m}{\Gamma(m+1)\Gamma(\theta_{on} + \theta_{off} + m)\Gamma(\theta_{on})}\ {}_1F_1(\theta_{on} + m, \theta_{on} + \theta_{off} + m, -\theta_t) \quad (2)$$

where ${}_1F_1$ represents the confluent hypergeometric function of the first kind.

Random dropout event model

If desired and when appropriate, users can account for random dropout events in CALISTA. This feature is particularly suitable when dealing with single-cell transcriptional profiles from Drop-Seq technology. Here, the dropout probability is modeled by a negative exponential function with an optimal decay constant of λ, as follows:

$$P_d(0|m) = e^{-\lambda m} \quad (3)$$

where $P(0|m)$ denotes the probability that the measured single-cell mRNA count is zero when the true number of mRNA molecules is *m*. A decay constant $\lambda = 0$ gives a dropout probability of 1, i.e. dropout occurs regardless of the true mRNA count. The parameter λ is estimated from the plot of the fraction of zeros against the mean expression across all measured genes, following the procedure described in Pierson et al. (35) (see Supplementary Figure S2). Combining the dropout event model with the two-state gene transcription model above give the following distribution function of measured single-cell mRNA readouts $\widehat{m}$:

$$P(\widehat{m};\ \theta_{on}, \theta_{off}, \theta_t) = \begin{cases} \sum_m P_d(0|m) P_2(m;\ \theta_{on}, \theta_{off}, \theta_t) & \text{for } \widehat{m} = 0 \\ \left(1 - e^{-\lambda \widehat{m}}\right) P_2(\widehat{m};\ \theta_{on}, \theta_{off}, \theta_t) & \text{for } \widehat{m} > 0 \end{cases} \quad (4)$$

Maximum likelihood clustering by greedy algorithm

As illustrated in Figure 1b, the greedy algorithm for the maximum likelihood single-cell clustering in CALISTA comprises three basic steps:

1. Initialization. Cells are randomly assigned into *K* clusters with equal probabilities (i.e., uniform random cell assignments).

2. Parameter estimation. For each cluster *k* (between 1 and *K*) and each gene *g* (between 1 and *G*), CALISTA obtains the parameter vector of the two-state gene transcription model $\theta^*(g,k) = \{\theta^*_{on}, \theta^*_{off}, \theta^*_t\}^k_g$ that maximizes the joint probability of obtaining the gene expression values of the cells in the cluster, as follows:

$$\theta^*(g,k) = arg\max_{\theta} \sum_{n \in N_k} \ln P(\hat{m}_{ng}; \theta) \quad (5)$$

where N_k is the set of cell indices for the cells in cluster *k.* The probability $P(\hat{m}; \theta)$ is computed using the steady state probability distribution from the two-state gene transcription model as given in Eq. (2) (assuming that random dropout events are insignificant) or using the distribution function defined in Eq. (4).

For computational speedup, as previously done in SABEC (21), the log-probability $\ln P(\hat{m}|\theta)$ values are pre-computed for *m* between 0 and 200 and for a set of parameter vectors $\{\theta^w_{on}, \theta^w_{off}, \theta^w_t,\}^W_{w=1}$, producing a $W \times 201$ matrix *L*. More specifically, the (*i, j*)-th element of *L* gives the log-probability $\ln P(j|\theta_i)$. The parameter set consists of all combinations from the following parameter discretization: θ_{on} =[0.5, 1, 2, 3, 4, 5], θ_{off} =[0.25, 0.5, 1, 2, 3, …, 19], and θ_t = [1, 51, 101, …, 451]. The parameter ranges in CALISTA are adapted from SABEC and selected to cover the parameter combinations for which $P(\hat{m}|\theta)$ spans a family of steady state distributions of mRNA counts, including negative binomial, Poisson, and bimodal distributions. While the number of parameter combinations (*W* = 945) in CALISTA is much lower than that in SABEC (*W* = 39500), our extensive testing indicated that the single-cell clustering in CALISTA performs as well as SABEC at much lower computational requirements.

By pre-computing the log-probability values, the solution to the parameter estimation problem above reduces to finding the parameter combination corresponding to the maximum element of the following vector:

$$L \times r^{k,g} \quad (6)$$

where $r^{k,g}$ is a 201×1 vector with its *i*-th element being the number of cells in cluster *k* with its expression value of gene *g* rounded to the nearest integer equaling to $\hat{m}_{n_k,g}$.

3. Cell re-assignment. Once the optimal parameter values for all *G* genes and all *K* clusters have been obtained, CALISTA reassigns each cell to its new cluster following a greedy algorithm. More specifically, CALISTA evaluates the likelihood for each cell to belong to a cluster as follows:

$$\Lambda_k(n) = \sum_{g=1}^{G} \ln P\left(\widehat{m}_{n,g}; \theta^*(g,k)\right) \quad (7)$$

where $\Lambda_k(n)$ denotes the likelihood value of the *n*-th cell to be in cluster *k*. We reassign each cell to the cluster that gives the maximum likelihood.

Steps 2 and 3 from the algorithm outlined above are repeated iteratively until convergence, i.e. until there are no new cell reassignments.

Single-cell clustering using consensus matrix

To improve both the robustness of the single-cell clustering, CALISTA executes the iterative greedy clustering algorithm above until convergence for a user-specified number of times (by default 50), each starting with a different random initial cell assignment. The results from the independent clustering runs are combined to produce a $N \times N$ symmetric consensus matrix C. The (*i,j*)-th element of the matrix *C* gives the number of times that the *i*-th and *j*-th cells are clustered together, as illustrated in Figure 1b. CALISTA subsequently applies *k*-medoids algorithm using the consensus matrix to obtain the final cell clustering assignment (33).

Choosing the number of clusters

As an aid for choosing the appropriate number of clusters *K* – when the number is not known *a priori* – CALISTA provides the eigengap plot based on the normalized graph Laplacian matrix $A_{N\times N}$ of the consensus matrix *C*, defined as follows (36):

$$A = I - B^{-1/2} C B^{-1/2} \quad (8)$$

where I is the $N \times N$ identity matrix and *B* is the $N \times N$ degree matrix of C, i.e. the diagonal matrix whose elements represent the row-wise sum of the consensus matrix (i.e., $b_{i,i} = \sum_{j=1}^{N} c_{i,j}$). The eigengap plot is the plot of the eigenvalues λ of the matrix *A* in an ascending order. The number of clusters *K* can be heuristically chosen based on a "gap" in the eigenvalues of *A*, where the lowest *K* eigenvalues $\lambda_{1\ldots}\ \lambda_K$ values are much smaller than λ_{K+1}.

Lineage Inference

The second main component of CALISTA is the reconstruction of cell lineage graph, which reflects the lineage progression in the differentiation process (Figure 1c). Based on the view that cell clusters represent cell states, the nodes of the lineage graph comprise cell clusters, while the edges represent state transitions in the lineage progression. For inferring the lineage graph, CALISTA computes cluster distances based on dissimilarities in the gene expressions among cells from two clusters. Again, CALISTA adopts a likelihood-based strategy using the probability distribution of mRNA from the stochastic two-state gene

transcription model to define the cluster distances. Finally, users can use CALISTA to extract the set of transition genes between any two connected clusters in the lineage graph.

Cluster distance

Given *K* clusters from the single-cell clustering analysis above (or provided by the user), CALISTA evaluates a $K \times K$ dissimilarity matrix S where the element s_{kj} gives the likelihood of cells from cluster *k* to be assigned to cluster *j*, computed as follows:

$$s_{kj} = \frac{1}{N_k}\frac{1}{G}\left(\sum_{n_k=1}^{N_k}\sum_{g=1}^{G}\ln P\left(\widehat{m}_{n_k,g};\theta^*(g,j)\right)\right) \quad (9)$$

Note that the diagonal element s_{kk} is the sum of the cell likelihood in the original cluster assignment, i.e. $s_{kk} = \sum_{n\in N_k}\Lambda_k(n)$, and should be larger than other elements s_{kj}, $j \neq k$. The dissimilarity coefficients in the matrix S is subsequently normalized by subtracting each element with the diagonal element of the corresponding rows, as follows:

$$\hat{s}_{kj} = s_{kk} - s_{kj} \quad (10)$$

Since the likelihood always takes on negative values, the normalized dissimilarity coefficient assumes a positive value. A larger $\hat{s}_{kj}$ reflects a higher degree of dissimilarity between the two clusters. The distance between the *j*-th and *k*-th cluster, denoted by d_{kj}, is defined as:

$$d_{kj} = max\left(\hat{s}_{kj}, \hat{s}_{jk}\right) \quad (11)$$

Lineage graph construction

CALISTA generates a lineage graph by connecting single-cell clusters (states) based on the cluster distances. The lineage graph describes the state transition of stem cells during the cell differentiation process, under the assumption that the transitions occur between closely related cell states (i.e. between clusters with low distances). Briefly, CALISTA starts with a fully disconnected graph of single cell clusters, and recursively adds one transition edge at a time in increasing magnitude of cluster distance until each cluster is connected by at least one edge. CALISTA further allows the users to manually add or remove transition edges between pairs of clusters/nodes based on the cluster distance ranking, time/cell stage information or biological knowledge. Once the lineage graph has been established, CALISTA assigns directionalities to the edges in the lineage graph according to user-provided information (e.g. stage or time information, starting cells, or expected gene expression profiles). When the stage or time information is available for the cells, the clusters are first labeled using the most frequent value (mode) of cells' stage or time. CALISTA then assigns

the edge directionalities according to the causal direction of the cluster stage or time. Alternatively, CALISTA can also employ users provided information on the start cell or a marker gene with a known expression profile during cell differentiation (e.g., in in Figure 1d, gene Nfe2 is expected to be downregulated during the differentiation process) to determine the starting cell cluster. More specifically, given the start cell, CALISTA defines its corresponding cluster as the initial cell state of the lineage graph, whereas based on a marker gene's profile, CALISTA calculates its mean expression for each cell cluster and identifies the starting cell state as the one with minimum (maximum) averaged value in case of upregulation (downregulation).

In the applications of CALISTA to previously published single-cell gene expression datasets, we followed a few guidelines in curating the lineage progression graph using the capture time or developmental stage information. First, we added transition edges in increasing cluster distances until all paths in the lineage progression begins with cluster(s) having the lowest capture time or stage (i.e., clusters with the majority of the cells coming from the first time point or stage). We subsequently pruned edges to eliminate shortcuts in the lineage progression that create redundant paths. For example, given three clusters of increasing cluster time – A, B, and C, respectively – and a lineage progression graph with transition edges A → B, B → C, and A → C, we thus remove the edge A → C as this path bypasses the cluster B.

Transition genes

Given an edge between cluster *k* and *j* in the lineage graph, CALISTA provides the set of transition genes whose expressions are differentially distributed between the two clusters. More specifically, CALISTA evaluates the likelihood difference between having cells in clusters *j* and *k* separately and having the cells together in one cluster, again using the steady-state distribution of mRNA from the two-state gene transcription model, as follows:

$$v_{jk}^{g} = v_{j}^{g} + v_{k}^{g} - v_{j+k}^{g} \qquad (12)$$

with

$$v_{j}^{g} = \sum_{n \in N_j} \ln P\left(\widehat{m}_{n,g}; \theta^{*}(g,j)\right) \qquad (13)$$

$$v_{k}^{g} = \sum_{n \in N_k} \ln P\left(\widehat{m}_{n,g}; \theta^{*}(g,k)\right) \qquad (14)$$

$$v_{j+k}^{g} = \sum_{n \in N_k \cup N_j} \ln P\left(\widehat{m}_{n,g}; \theta^{*}(g,j+k)\right) \qquad (15)$$

where the optimal parameter vector $\theta^{*}(g, j+k)$ is obtained according to Equation (5) for the combined cells in clusters *j* and *k*. A large v_{jk}^{g} reflects that gene *g* is differentially expressed

between clusters *j* and *k*, according to the stochastic two-state gene transcriptional model. For each edge in the lineage graph, CALISTA generates a rank list of genes in decreasing values of v_{jk}^{g} . The transition genes correspond to the set of top genes in the list such that the ratio between the sum of v_{jk}^{g} among these genes and the total sum of v_{jk}^{g} among all genes exceeds a given threshold (default threshold: 50%).

Pseudotemporal ordering of cells

Given a lineage progression graph among cell clusters, the third and last main component of CALISTA concerns the pseudotemporal ordering of cells. For this purpose, we first set cluster pseudotimes. If the time or stage information of the cells is provided, then the pseudotime of each cluster is set to the mode of the time/stage of the cells in the cluster divided by the largest time/stage. When the time/stage information is not available, but the starting cluster is known (e.g., from knowledge of starting cells or marker genes), we assigned a pseudotime of 0 for the starting cluster. We then evaluated the sum of the cluster distances along each path in the lineage progression and identified the maximum cumulative cluster distance. The pseudotime of a cluster is given by its cumulative cluster distance to the starting cluster divided by the maximum cumulative cluster distance. CALISA also allows the user to manually assign the pseudotimes of the clusters. Once the cluster pseudotimes have been set, we assign each cell to one of the state transition edges and a pseudotime using the maximum likelihood principle (see below). Finally, given a developmental path in the linage progression, CALISTA provides a pseudotemporal ordering of cells along the defined path.

Cell assignment to transition edges

For pseudotemporal ordering of the cells, CALISTA first assigns cells to the edges in the lineage graph. In the following illustration, let us consider a cell *n* in cluster *k.* CALISTA allocates the cell *n* to one of the edges that are incident to cluster *k*, again following the maximum likelihood principle. For this purpose, we define the likelihood value of a cell *n* to belong to an edge pointing from any cluster *j* to cluster *k* as follows:

$$\Lambda_{j\to k}(n) = \max_{t} \Lambda_j(n) + \frac{(t - t_j)}{(t_k - t_j)}\left(\Lambda_k(n) - \Lambda_j(n)\right) \quad (16)$$

where t_k denotes the cluster pseudotime label. Similarly, we define the likelihood value of the same cell to belong to an edge pointing from cluster *k* to any cluster *l* by the following:

$$\Lambda_{k\to l}(n) = \max_{t} \Lambda_k(n) + \frac{(t - t_k)}{(t_l - t_k)}\left(\Lambda_l(n) - \Lambda_k(n)\right) \quad (17)$$

CALISTA computes all possible $\Lambda_{j\to k}(n)$ and $\Lambda_{k\to l}(n)$ lineage graph, and assigns the cell to the edge that gives the maximum of all $\Lambda_{j\to k}(n)$ and $\Lambda_{k\to l}(n)$ values. The pseudotime of the cell $t(n)$ is set to *t* that gives the maximum likelihood value, as follows:

$$t(n) = \arg\max_t \Lambda_j(n) + \frac{(t - t_j)}{(t_k - t_j)}\left(\Lambda_k(n) - \Lambda_j(n)\right)$$
$$or \qquad (18)$$
$$t(n) = \arg\max_t \Lambda_k(n) + \frac{(t - t_k)}{(t_l - t_k)}\left(\Lambda_l(n) - \Lambda_k(n)\right)$$

depending on the cell assignment to edges above.

Cell ordering along a developmental path

Given a lineage progression graph, users can identify one or several developmental paths. A developmental path is defined as the sequence of connected clusters in the lineage progression graph with transition edges pointing from one cluster to the next in the sequence. CALISTA generates a pseudotemporal ordering along a given developmental path by first identifying cells belonging to the state transition edges in the path and order these cells according to their pseudotimes. Note that in defining the likelihood function for assigning cells to edges, we have assumed that the steady state probability distributions of gene expressions vary linearly between two connected clusters or states. But the result of the cell ordering does not change if we replace the linear varying function in the likelihood definition with any monotonic function.

Single-cell Analysis of Zebrafish Embryogenesis

In the clustering analysis of scDrop-seq from zebrafish embryogenesis (37), we applied a modified procedure as follows:

1. Cluster cells from the final developmental stage / time *t.* The optimal number of clusters K_t was chosen as the maximum *K* among the top three eigengaps (see above and Supplementary Figure S3).
2. Cluster cells from the previous stage *t* – 1 with the following constraint on the number of clusters:
3. $$K_{t-1} \in \begin{cases} \left[\max\left(1, \left\lfloor \frac{K_t}{3} \right\rfloor\right), K_t\right] & if\ t > 3 \\ [1, K_t] & if\ t \le 3 \end{cases} \qquad (19)$$

   CALISTA automatically finds the optimal number of clusters K_{t-1} within the permitted range again as the maximum *K* among the top three eigengaps (see above and Supplementary Figure S3).
4. Repeat step 2 until the data from all time points are analyzed.

After cell clustering, CALISTA selects the most informative genes as the union of the most variable genes detected at each time point and re-scales the original data accordingly (see Input data and data proprocessing section above). For reconstructing the lineage progression, we considered transitions or edges only between clusters of adjacent time points, again in the order of increasing cluster distances. We added a transition edge such that each cluster has one incoming and outgoing edge, except for the initial and final clusters that have only one in out-edge and in-edges, respectively.

RESULTS

Single-cell Transcriptional Analysis using CALISTA

Figure 1 summarizes the four analyses of single-cell transcriptional profiles in CALISTA, including: (1) clustering of cells, (2) reconstruction of cell lineage progression (3) identification of key transition genes, and (4) pseudotemporal ordering of cells. In CALISTA, we adopt a likelihood-based approach where the likelihood of a cell is computed using a probability distribution of mRNA defined according to the two-state model of gene transcriptional process (34) and when appropriate, a random dropout event model (see Material and methods). A random dropout occurs when mRNA molecules of a gene are not detected even though the true mRNA count is non-zero. As illustrated in Figure 1b, the single-cell clustering in CALISTA involves two steps: (1) independent runs of maximum likelihood clustering using a numerically efficient greedy algorithm, and (2) consensus clustering using *k*-medoids (33). The maximum likelihood clustering in the first step starts with a random assignment of cells into a desired number of clusters, followed by iterations of (a) maximum likelihood estimation of two-state model parameters for each gene and each cluster and (b) reassignments of each cell to the cluster that give the largest cell likelihood value, until no more cell reassignments occur or until a prescribed maximum number of iterations is reached. A consensus matrix containing the number of times two cells are assigned to the same cluster is generated from repeated runs of the maximum likelihood clustering. We employ *k*-medoids to obtain the final cluster using the consensus matrix. The single-cell clustering in CALISTA is similar to a previously published clustering algorithm SABEC (21), but offers a substantial improvement in computational speed due to the implementation of a greedy algorithm in CALISTA (see Supplementary Table S2 for comparison of computational times). The computational efficiency of CALISTA is important especially for scaling up the analysis to extremely large single-cell transcriptomics datasets with tens to hundreds of thousands of cells, such as those generated by scDrop-Seq technology.

By viewing the single-cell clusters as cell states through which stem cells transition during the cell differentiation, CALISTA uses distances between any pair of cell clusters to reconstruct the cell lineage progression. The cluster distance – defined as the maximum difference in the cumulative likelihood value upon reassigning the cells from the original cluster to the other cluster (see Material and methods) – gives a measure of dissimilarity in their gene expression distributions between any two clusters. CALISTA generates the lineage

progression graph by sequentially adding state transition edges connecting closely distanced clusters until every cell cluster is connected to at least another cluster (Figure 1c). Notably, CALISTA provides an interface for users to curate the lineage progression graph by manually adding or removing state transition edges, guided by the cluster distances and other available information about the cell differentiation. For assigning directionality to the edges, CALISTA relies on user-provided information, for example information on the cell stage or sampling time, the starter/progenitor cells or the expected temporal profiles of the gene expression.

For any two connected clusters in the lineage progression, one can further obtain the set of transition genes using CALISTA. The transition genes are determined based on gene-wise likelihood differences between having the cells in separate clusters and having them together in a single cluster (see Material and methods). Here, the gene-wise likelihood difference reflects the informative power of a gene to segregate cells into two clusters. The transition genes may point to candidate gene markers and genes regulating the state transition during differentiation.

The final component of CALISTA concerns with the pseudotemporal ordering of cells along a developmental path – defined as a sequence of connected clusters and state transition edges – in the lineage progression graph (Figure 1d). Given a developmental path, CALISTA produces a list of the cells ordered in increasing pseudotimes. For this purpose, we first assign a pseudotime for each cluster, a normalized value such that the starting cluster in the lineage progression graph has a pseudotime of 0 and the final cell cluster (or clusters) has a pseudotime of 1 (see Material and methods). Subsequently, we assign each cell to a transition edge that is pointing to or emanating from the cluster to which the cell belongs, again by adopting the maximum likelihood principle. Here, we assume that the distribution of the single-cell gene expression varies monotonically between cell states (clusters). For simplicity, the likelihood of a cell along a transition edge is computed using a linear interpolation of the cell likelihood values for the connected clusters. Each cell is then assigned to the transition edge that maximizes its likelihood value. Analogously, the cell pseudotime is computed by a linear interpolation of the cluster pseudotimes and set to the corresponding maximum point of the cell likelihood value.

Application to the differentiation of induced pluripotent stem cells to cardiomyocytes

In the following, we demonstrated the application of the four analyses in CALISTA as a bioinformatics pipeline using the single-cell gene expression dataset from the differentiation of human induced pluripotent stem cells (iPSCs) into cardiomyocytes (26). The dataset includes single-cell expression of 96 genes measured by RT-qPCR for 1896 cells collected across 8 time points (day 0, 1, 1.5, 2, 2.5, 3, 4, 5) after induction to differentiate. Figure 2a shows the developmental states identified in the original study: epiblast cells (E) in the early stage (day 0, 1, 1.5), primitive streak (PS)-like progenitor cells in the intermediate stage (day 2, 2.5), and a lineage bifurcation into either a desired mesodermal (M) or an undesired endodermal (En) fate in the late stage (day 3, 4, 5).

First, we clustered the cells by using *k*-means algorithm and CALISTA. The optimal number of clusters was chosen to be five based on the eigengap plot (see Material and methods and Supplementary Figure S4). As depicted in Figure 2b, the single-cell clustering by *k*-means is incongruent with the known developmental states and the capture times. One of the cell clusters (cluster 3 in Figure 2b) includes cells from both early (E) and late stage (En and M) cells. On the other hand, the single-cell clustering result of CALISTA, as shown in Figure 2c, recapitulates the previously identified developmental states. Here, clusters 1, 2, and 5 contain mostly E, PS, and En cells, respectively, while M cells are split between clusters 3 and 4 that are demarcated by different capture times (see cluster compositions in Supplementary Figure S5).

After single-cell clustering, we employed CALISTA to infer the lineage progression graph (see Figure 2d). An edge connecting cluster 1 and 5 was manually removed as the edge was inconsistent with the cell capture time information (bypassing intermediate capture time points). The cluster pseudotimes were set to the modes (most frequent values) of the cell capture times in the clusters divided by the maximum cell capture time. The directionality of the state transition edges was finally set according to the cluster pseudotimes (also see Supplementary Figure S6), pointing from lower to higher pseudotimes. As illustrated in Figure 2d, the lineage progression graph reconstructed by CALISTA reproduces the lineage bifurcation event as the cells transition from PS-like cells to take on either M or En cell fates (26). Subsequently, for each state transition edge, we identified the set of transition genes (see Supplementary Figure S7). Among the identified transition genes in the lineage progression graph (25 genes in total), many are known lineage specific transcriptional regulators involved in the iPSC cell differentiation, such as EOMES, GATA4, GSC, HAND1, KIT, MESP1, SOX17 and T (26).

Finally, we employed CALISTA to generate the pseudotemporal ordering of cells along the two distinct developmental paths in the lineage progression: (1) the M path forming mesodermal cells (cluster 1 – 2 – 3 – 4; see red dashed arrow in Figure 2e) and (2) the En path forming endodermal cells (cluster 1 – 2 – 5; see green dashed arrow in Figure 2e). After assigning cells to the state transition edges and prescribing the cell pseudotimes, we ordered cells belonging to each developmental trajectory in increasing pseudotimes with a total of 1408 cells in the M path and 1215 cells in the En path.

The likelihood value of each cell computed during the pseudotemporal cell ordering can further be visualized as a landscape plot. Figure 2f depicts the negative log-likelihood surface of the cells over the first two principal components. A higher value on the surface indicates a cell state with broader mRNA distributions, i.e. a state of higher uncertainty in the gene expression. As depicted in Figure 2f, iPSC cells start their journey from a valley in this surface, implying that the progenitor cells are at a low uncertainty state. As the cell differentiation progresses, cells pass through an intermediate state with higher uncertainty, where a peak uncertainty is reached at or around the cell lineage bifurcation. After the bifurcation, cells follow two paths toward lower uncertainty, leading to two valleys

corresponding to distinct cell fates (M and En fates). The rise-and-fall in gene expression uncertainty have also been reported in other cell differentiation systems, suggesting that stem cells go through a transition state of high uncertainty before committing to their final cell fate(s) (7, 12).

To visualize the pseudotemporal trajectories of the gene expression along the two cell differentiation paths, we calculated the moving average expression values of transition genes for the pseudotemporally ordered cells using a moving window comprising 10% of the total cells in each path (see Figure 2g-i and Supplementary Figure S8). A number of transition genes follow highly similar expression trajectories along the M and En paths, with an increase in expression from E to PS-like state, followed by a decrease in expression from PS-like to M or En state (see Figure 2g). The majority of genes with the aforementioned trajectory are known PS-like markers (for example EOMES, GSC, MESP1, and MIXL1 (26, 38, 39)), and thus, we refer to these genes as PS-genes (see Supplementary Table S3). Another group of transition genes shows differential profiles between the M and En paths after the lineage bifurcation. Here, we define M-genes as the genes with higher expressions along the M path than the En path (see Figure 2h). Correspondingly, we define En-genes as genes with higher expressions along the En path than the M path (see Figure 2i). Notably, many of the known M marker genes (e.g., BMP4, HAND1, MYL4 and TNNT2 (26, 40)) are among the M-genes, and several of the known En marker genes (e.g. HNFA4, KIT, and SOX17 (26, 41, 42)) are among the En-genes (see Additional file 3). In general, after the lineage bifurcation, the expressions of M genes increase along the M path, but are either suppressed (BMP4, HAND1, KDR) or unchanged (MYL4, TGFB2, TNNT2) along the En path. Among the known En markers, the expression of HNFA4 increases along the En path after the lineage bifurcation but remains relatively unchanged along the M path. The expression of KIT follows the opposite profile, where the gene expression is downregulated along the M path and is relatively unchanged along the En path. While the expression of SOX17, like KIT, decreases along the M path, the gene is upregulated along the En path immediately after the lineage bifurcation before being downregulated toward the end of the developmental trajectory.

Finally, we constructed gene co-expression networks for the M and En developmental paths based on the pseudotemporal profiles of the gene expression (pairwise Pearson correlation, p-value ≤ 0.01 and correlation value ≥ 0.8, see Supplementary Figure S9). We identified cliques in the M and En gene co-expression networks, i.e. a subset of genes (at least 5) that are connected to each other, using a maximal clique analysis by the Bron-Kerbosch algorithm (43). Figure 2j depicts the cliques from the M and En gene co-expression networks, showing three gene regulatory modules: one module specific to the M path (red), another specific to the En path (green), and a shared module (grey). Most of the PS genes, including known PS marker genes, belong to the shared module as expected. Among the genes in the shared module, DKK1, FZD1 and T are involved in Wnt signaling pathway, which is known to promote cell differentiation (44). GATA6, which plays an important role in the endoderm commitment (38), belongs to the En module. On the other hand, the M module

shows activating relationships among mesoderm gene markers (e.g. BMP4, MYL4 and HAND1), and antagonistic interactions between several M genes and KIT, an En gene and marker.

Comparison to other material and methods

Single-cell clustering

We compared the performance of CALISTA to three popular single-cell clustering algorithms, namely MONOCLE 2(24), SC3(19) and SIMLR(20). SC3 and SIMLR are consensus-clustering material and methods that learn cell-cell similarities by using multiple kernels and Laplacian transformations, respectively. MONOCLE 2 is designed for single-cell gene expression data analysis that also includes, among its features, a single-cell clustering algorithm using a *t*-distributed method. For performance comparison, we applied each of the clustering material and methods above and CALISTA to four publicly available scRNA-seq datasets (45–48), including Kolodziejczyk et al. study on the differentiation of pluripotent stem cells under different conditions (45), Pollen et al. study on the identification of different cell types within a heterogeneous human fetal population (46), Usoskin et al. study using single mouse neural cells (47), and Villani et al. study on the detection of human blood dendritic cell subpopulations (48). To assess the clustering accuracy of each method, we calculated the Adjusted Rand Index (ARI) (19, 49). An ARI of 0 indicates fully incorrect labels, while a value of 1 indicates fully correct labels. The reference cell labels were taken from the original studies. The results in Table 1 show all of the clustering algorithms perform comparably well for two of the four datasets, namely Kolodziejczyk et al. (45) and Pollen et al. (46) datasets, with ARI values exceeding 0.85. But, for the other two datasets (Usoskin (47) and Villani (48) datasets), CALISTA and SC3 outperform SIMLR and MONOCLE 2 (see Supplementary Table S4 for clustering results).

Lineage inference and pseudotemporal ordering

We further compared the performance of CALISTA in reconstructing the lineage progression with three existing algorithms that are able to infer linear and branched trajectories, namely Diffusion MONOCLE 2 (24), Pseudotime (DPT) (22) and SCUBA (17). MONOCLE 2 and SCUBA are able to detect multiple bifurcation events in the cell lineage progression, while DPT is able to identify only a single branching point (22). Of note, like CALISTA, no user input is required when reconstructing cell lineage progression using MONOCLE 2. In DPT, users have to specify information on the topology of the lineage progression (linear or branched), while SCUBA requires the cell stage information.

For performance comparison, we used four single-cell gene transcriptional datasets from cell differentiation systems with a variety of lineage topologies, including Bargaje et al. study above [11], Chu et al. study (50) on the differentiation of human embryonic stem cells into endodermal cells, Moignard et al. study (51) on hematopoietic stem cell differentiation, and Treutlein et al. study (11) on mouse embryonic fibroblast differentiation into neurons. Figure 3

shows the reconstruction of lineage progression from single-cell data of Chu et al., Moignard et al and Treutlein et al. study using CALISTA (see Supplementary Note S2 and Figure S10-11 for more details). The lineage progressions reconstructed using MONOCLE 2, DPT and SCUBA algorithms are provided in Supplementary Figures S12-S14, respectively.

Table 2 compares CALISTA and the above algorithms in reconstructing cell lineage progression. For each of the datasets, CALISTA is able to produce clustering results that are consistent with the known population substructures, and to accurately identify single or multiple bifurcation events. MONOCLE 2 is able to correctly detect single bifurcation event in Bargaje et al. and Treutlein et al. dataset, but only one of the two bifurcation events in Moignard et al. study (Supplementary Figure S12). For Chu et al. dataset, MONOCLE 2 identifies multiple bifurcation events when none is expected. Meanwhile, DPT reconstructs lineage progression graphs of Bargaje et al. and Chu et al. datasets that are incongruent with the differentiation paths according to the cell capture times (see Supplementary Figure S13). For Moignard et al. dataset, DPT inferred lineage progression is in general agreement with the known differentiation stages, but because of the limitation mentioned above, DPT is able to detect only one of the two bifurcation events. For Treutlein et al. dataset, DPT accurately reproduces the known lineage progression. Finally, SCUBA is able to recover single bifurcation events in the lineage progression of Bargaje et al. and Treutlein et al. dataset, but mistakenly identifies a single bifurcation event for Chu et al. dataset (see Supplementary Figure S14). We were not able to use SCUBA to analyze the dataset of Moignard et al. study (51), since the capture time information was not available.

Application of CALISTA to massively parallel Drop-Seq datasets

To demonstrate the scalability of CALISTA, we analyzed single-cell expression datasets from droplet-based assays. Single-cell Drop-seq is a massively parallel genome-wide expression profiling technology capable of analyzing thousands of cells in a single experiment. However, the bioinformatic analysis of large single-cell transcriptomics datasets is computationally challenging (52). CALISTA offers a parallelization option for handling large datasets by splitting the greedy optimization runs among multiple computing cores.

We tested the single-cell clustering performance of CALISTA in analyzing Drop-seq datasets using the single-cell study of mouse spinal cord neurons by Sathyamurthy et al. (~18K nuclei) (53) and the study of peripherical blood mononuclear cells by Zheng et al. (~68K cells) (31). We noted that *k*-medoid clustering on very large consensus matrices is computationally prohibitive. Thus, we bypassed the consensus clustering and took the cell clustering corresponding to the highest likelihood value among the independent runs of the greedy algorithm. For Sathyamurthy et al. dataset, CALISTA clustering converged to nine single-cell clusters in good agreement with the original study (53) (see Supplementary Figure S15 and Note S3). By using 29 marker genes (53), CALISTA is able to further split the large cluster of neurons (52% of total data (53)) into two cell subpopulations – one comprising cells with high expressions of Snap25, Syp and Rbiox3, and the other comprising cells with

medium expression of these genes (see Supplementary Figure S15). Notably, CALISTA clustering analysis is able to distinguish Schwann and Meningeal subpopulations of cells, whose differences are undetected in the original study (53). For Zheng et al. scDrop-seq dataset, CALISTA converged to ten clusters, also in general agreement with the original study (see Supplementary Figure S16).

We further tested CALISTA lineage progression inference using scDrop-seq data of ~38K cells from 12 developmental stages of zebrafish embryogenesis (37), the results of which are illustrated in Figure 4. For this extremely large dataset, we employed a modified clustering procedure (see Material and methods) and identified 82 clusters, of which 23 comprise cells at the final developmental stage (see Figure 4a-b and Supplementary Note S3). Figure 4b-c show single-cell clustering and the reconstructed lineage progression by CALISTA, respectively (see also Supplementary File S1). Figure 4d shows the cell type labels for the cell clusters at the final developmental stage based on the expression levels of 26 key gene markers (37) (see Supplementary Figure S17). The results are again in good agreement with the known cell types (37).

Algorithmic stability

For assessing the robustness of CALISTA, we reran the CALISTA clustering analysis for random subsamples of Bargaje et al. (26), Chu et al. (50), Moignard et al. (51), and Treutlein et al. (11) datasets, using the same settings employed for the full datasets. We calculated the Figure of Merit (FOM) (54), the area under the Receiver Operating Characteristic (AUROC), and the Precision-Recall curve (AUPR) (55) to judge the similarity between the single-cell clustering results of the data subsamples and the full datasets. The comparisons show that CALISTA single-cell clustering results of data subsamples are highly congruent with those of the full dataset across different percentages of data subsampling, from 90% down to 60% of the full dataset (see Supplementary Figure S18 and Table S5).

Computational complexity

The most computationally expensive step in CALISTA is the single-cell clustering analysis. To assess the scalability and computational complexity of CALISTA clustering, we recorded the computational runtimes for nine of the expression single-cell transcriptomics datasets analyzed above (11, 26, 31, 37, 46, 48, 51, 53), covering a range from ~250 to 68K cells. The computational times show that for datasets with <10k cells, the single-cell clustering analysis typically takes less than one minute on a standard workstation with 3.5GHz Intel Xeon processor (see Supplementary Figure S19). For larger datasets, although the computational times expectedly increase, the increase is linear with respect to the number of cells. Furthermore, the independent runs of the greedy algorithm are embarrassingly parallel and thus, can be easily and efficiently implemented across multiple computing cores (see Supplementary Figure S19).

DISCUSSION

CALISTA provides four bioinformatic tools for single-cell expression data analysis that are essential in single-cell studies of stem cell differentiation. These analyses can be implemented either sequentially in a data analytics pipeline or separately in a standalone application. The analytical tools in CALISTA are developed using the same likelihood-based approach and therefore are fully compatible with each other. To the best of our knowledge, CALISTA is the first mechanistic model-based toolbox that allows an end-to-end analysis of single-cell transcriptional profiles. Despite the focus on stem cells in our work, each of the tools in CALISTA is agnostic to the source of the single-cell transcriptomic data analysis and thus can be used for other cellular systems.

In comparison to SABEC (21), another clustering algorithm that also uses the stochastic dynamic model of gene transcriptional bursts, the clustering analysis in CALISTA offers a significant advantage in terms of computational efficiency thanks to the implementation of greedy algorithm. This advantage is critical and necessary for the analysis of large datasets with tens to hundreds of thousands of cells, such as those generated by Drop-seq technology. In addition to computational efficiency, CALISTA is able to account for random technical dropout events, a feature that is missing from SABEC. Finally, and importantly, CALISTA permits other types of analysis beside single-cell clustering, for which SABEC is solely developed. As demonstrated in numerous case studies above, when compared to popular single-cell analysis material and methods, many of which were developed for a specific type of single-cell data analysis (e.g. clustering, pseudotemporal ordering), CALISTA performs as well as, if not better than, these algorithms.

CALISTA further features an interactive interface for user inputs at different steps in the analysis, for example in setting the number of cell clusters or during the curation of the lineage progression graph (see CALISTA user manual on https://github.com/CABSEL/CALISTA). The user interface enables incorporating existing biological knowledge of the cell differentiation system, which is often difficult – if not impossible – to codify. Although such prior knowledge is not necessary for using CALISTA, the ability to incorporate this knowledge, whenever available, is a useful and important feature in the analysis of single-cell transcriptional profiles.

CONCLUSION

A key challenge in analyzing single-cell transcriptional profiles is the high biological and technical noise. Because gene transcriptions occur in stochastic bursts, cells at the same biological state can naturally display different gene expression profiles (34). Random technical dropouts add noise to single-cell mRNA counts, leading to zero-inflated distributions and further complicating data analysis. In this work, we developed CALISTA, a highly scalable and efficacious toolbox for the analysis of single-cell gene expression data. We addressed the challenge of high data noise by adopting a likelihood-based approach based on a combination of mechanistic models of stochastic bursty gene transcriptional process and

random dropout events. CALISTA includes essential tools to carry out an end-to-end analysis of single-cell transcriptional profiles, including (1) single-cell clustering, (2) lineage progression inference, (3) transition genes identification, and (4) pseudotemporal ordering of cells. As demonstrated in numerous case studies, CALISTA performs as well as, if not better than, existing algorithms that are developed and optimized to carry out a particular type of single-cell analysis. Finally, as all tools in CALISTA share the same underlying likelihood-based principle, the different analyses within CALISTA are fully compatible with each other.

AVAILABILITY

A MATLAB and R version of CALISTA used in this study is freely available from the following website: https://github.com/CABSEL/CALISTA.

All the public single cell data sets analyzed in this study are available from the original publications. *In silico* datasets for evaluating the effects of gene expression scaling are available on CALISTA website: https://github.com/CABSEL/CALISTA.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR online.

FUNDING

This work was supported by the Swiss National Science Foundation (grant number 157154 and 176279).

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

REFERENCES

1. Ralston,A. and Shaw,K. (2008) Gene Expression Regulates Cell Differentiation. *Nat. Educ.*, **1(1)**, 127.
2. Kalisky,T., Oriel,S., Bar-Lev,T.H., Ben-Haim,N., Trink,A., Wineberg,Y., Kanter,I., Gilad,S. and Pyne,S. (2018) A brief review of single-cell transcriptomic technologies. *Brief. Funct. Genomics*, **17**, 64–76.
3. Guo,G., Huss,M., Tong,G.Q., Wang,C., Li Sun,L., Clarke,N.D. and Robson,P. (2010) Resolution of cell fate decisions revealed by single-cell gene expression analysis from zygote to blastocyst. *Dev. Cell*, **18**, 675–85.
4. Kumar,R.M., Cahan,P., Shalek,A.K., Satija,R., DaleyKeyser,A.J., Li,H., Zhang,J., Pardee,K., Gennert,D., Trombetta,J.J., *et al.* (2014) Deconstructing transcriptional heterogeneity in pluripotent stem cells. *Nature*, **516**, 56–61.
5. Cacchiarelli,D., Trapnell,C., Ziller,M.J., Soumillon,M., Cesana,M., Karnik,R., Donaghey,J., Smith,Z.D., Ratanasirintrawoot,S., Zhang,X., *et al.* (2015) Integrative Analyses of Human Reprogramming Reveal Dynamic Nature of Induced Pluripotency. *Cell*, **162**,

412–424.
6. Petropoulos,S., Edsgä,D., Linnarsson,S., Sandberg,R., Lanner,F., Rn Reinius,B., Deng,Q., Panula,S.P., Codeluppi,S. and Reyes,A.P. (2016) Single-Cell RNA-Seq Reveals Lineage and X Chromosome Dynamics in Human Preimplantation Embryos. *Cell*, **165**.
7. Richard,A., Boullu,L., Herbach,U., Bonnafoux,A., Morin,V., Vallin,E., Guillemin,A., Papili Gao,N., Gunawan,R., Cosette,J., *et al.* (2016) Single-Cell-Based Analysis Highlights a Surge in Cell-to-Cell Molecular Variability Preceding Irreversible Commitment in a Differentiation Process. *PLOS Biol.*, **14**, e1002585.
8. Kærn,M., Elston,T.C., Blake,W.J. and Collins,J.J. (2005) Stochasticity in gene expression: from theories to phenotypes. *Nat. Rev. Genet.*, **6**, 451–464.
9. Raj,A., Peskin,C.S., Tranchina,D., Vargas,D.Y., Tyagi,S., Becskei,A., Kaufmann,B., Oudenaarden,A. van, Blake,W., Kaern,M., *et al.* (2006) Stochastic mRNA Synthesis in Mammalian Cells. *PLoS Biol.*, **4**, e309.
10. Grün,D., Lyubimova,A., Kester,L., Wiebrands,K., Basak,O., Sasaki,N., Clevers,H. and van Oudenaarden,A. (2015) Single-cell messenger RNA sequencing reveals rare intestinal cell types. *Nature*, 10.1038/nature14966.
11. Treutlein,B., Lee,Q.Y., Camp,J.G., Mall,M., Koh,W., Shariati,S.A.M., Sim,S., Neff,N.F., Skotheim,J.M., Wernig,M., *et al.* (2016) Dissecting direct reprogramming from fibroblast to neuron using single-cell RNA-seq. *Nature*, **534**, 391–395.
12. Stumpf,P.S., Smith,R.C.G., Lenz,M., Schuppert,A., Müller,F.-J., Babtie,A., Chan,T.E., Stumpf,M.P.H., Please,C.P., Howison,S.D., *et al.* (2017) Stem Cell Differentiation as a Non-Markov Stochastic Process. *Cell Syst.*, **5**, 268–282.e7.
13. Lin,P., Troup,M. and Ho,J.W.K. (2017) CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-seq data. *Genome Biol.*, **18**, 59.
14. žurauskienė,J. and Yau,C. (2016) pcaReduce: hierarchical clustering of single cell transcriptional profiles. *BMC Bioinformatics*, **17**, 140.
15. Buettner,F., Natarajan,K.N., Casale,F.P., Proserpio,V., Scialdone,A., Theis,F.J., Teichmann,S.A., Marioni,J.C. and Stegle,O. (2015) Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells. *Nat. Biotechnol.*, **33**, 155–160.
16. Xu,C. and Su,Z. (2015) Identification of cell types from single-cell transcriptomes using a novel clustering method. *Bioinformatics*, **31**, 1974–80.
17. Marco,E., Karp,R.L., Guo,G., Robson,P., Hart,A.H., Trippa,L. and Yuan,G. (2014) Bifurcation analysis of single-cell gene expression data reveals epigenetic landscape. *Proc. Natl. Acad. Sci. U. S. A.*, **111**, 5643–50.
18. Huang,W., Cao,X., Biase,F.H., Yu,P. and Zhong,S. (2014) Time-variant clustering model for understanding cell fate decisions. *Proc. Natl. Acad. Sci. U. S. A.*, **111**, E4797-806.
19. Kiselev,V.Y., Kirschner,K., Schaub,M.T., Andrews,T., Yiu,A., Chandra,T., Natarajan,K.N., Reik,W., Barahona,M., Green,A.R., *et al.* (2017) SC3: consensus clustering of single-cell RNA-seq data. *Nat. Methods*, **14**, 483–486.

20. Wang,B., Zhu,J., Pierson,E., Ramazzotti,D. and Batzoglou,S. (2017) Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning. *Nat. Publ. Gr.*, **14**.
21. Ezer,D., Moignard,V., Göttgens,B., Adryan,B., Ko,M., Nakauchi,H., Takahashi,N., Yu,J., Xiao,J., Ren,X., *et al.* (2016) Determining Physical Mechanisms of Gene Expression Regulation from Single Cell Gene Expression Data. *PLOS Comput. Biol.*, **12**, e1005072.
22. Haghverdi,L., Büttner,M., Wolf,F.A., Buettner,F. and Theis,F.J. (2016) Diffusion pseudotime robustly reconstructs lineage branching. *Nat. Methods*, **13**, 845–848.
23. Trapnell,C., Cacchiarelli,D., Grimsby,J., Pokharel,P., Li,S., Morse,M., Lennon,N.J., Livak,K.J., Mikkelsen,T.S. and Rinn,J.L. (2014) The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nat. Biotechnol.*, **32**, 381–6.
24. Qiu,X., Mao,Q., Tang,Y., Wang,L., Chawla,R., Pliner,H.A. and Trapnell,C. (2017) Reversed graph embedding resolves complex single-cell trajectories. *Nat. Methods*, **14**, 979–982.
25. Cannoodt,R., Saelens,W. and Saeys,Y. (2016) Computational methods for trajectory inference from single-cell transcriptomics. *Eur. J. Immunol.*, **46**, 2496–2506.
26. Bargaje,R., Trachana,K., Shelton,M.N., McGinnis,C.S., Zhou,J.X., Chadick,C., Cook,S., Cavanaugh,C., Huang,S. and Hood,L. (2017) Cell population structure prior to bifurcation predicts efficiency of directed differentiation in human induced pluripotent cells. *Proc. Natl. Acad. Sci. U. S. A.*, **114**, 2271–2276.
27. Vallejos,C.A., Risso,D., Scialdone,A., Dudoit,S. and Marioni,J.C. (2017) Normalizing single-cell RNA sequencing data: challenges and opportunities. *Nat. Methods*, **14**, 565–571.
28. Ståhlberg,A., Rusnakova,V., Forootan,A., Anderova,M. and Kubista,M. (2013) RT-qPCR work-flow for single-cell data analysis. *Methods*, **59**, 80–88.
29. Qiu,X., Hill,A., Packer,J., Lin,D., Ma,Y.-A. and Trapnell,C. (2017) Single-cell mRNA quantification and differential analysis with Census. *Nat. Methods*, **14**, 309–315.
30. Haghverdi,L., Lun,A.T.L., Morgan,M.D. and Marioni,J.C. (2018) Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nat. Biotechnol.*, **36**, 421–427.
31. Zheng,G.X.Y., Terry,J.M., Belgrader,P., Ryvkin,P., Bent,Z.W., Wilson,R., Ziraldo,S.B., Wheeler,T.D., McDermott,G.P., Zhu,J., *et al.* (2017) Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.*, **8**, 14049.
32. Macosko,E.Z., Basu,A., Satija,R., Nemesh,J., Shekhar,K., Goldman,M., Tirosh,I., Bialas,A.R., Kamitaki,N., Martersteck,E.M., *et al.* (2015) Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets. *Cell*, **161**, 1202–1214.
33. Bhat,A. (2014) K-MEDOIDS CLUSTERING USING PARTITIONING AROUND MEDOIDS FOR PERFORMING FACE RECOGNITION. *Int. J. Soft Comput. Math. Control*, **3**.
34. Peccoud,J. and Ycart,B. (1995) Markovian Modeling of Gene-Product Synthesis. *Theor.*

Popul. Biol., **48**, 222–234.
35. Pierson,E., Yau,C., Shapiro,E., Biezuner,T., Linnarsson,S., Blainey,P., Quake,S., Buettner,F., Natarajan,K., Casale,F., *et al.* (2015) ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biol.*, **16**, 241.
36. von Luxburg,U. (2007) A tutorial on spectral clustering. *Stat. Comput.*, **17**, 395–416.
37. Farrell,J.A., Wang,Y., Riesenfeld,S.J., Shekhar,K., Regev,A. and Schier,A.F. (2018) Single-cell reconstruction of developmental trajectories during zebrafish embryogenesis. *Science*, **360**, eaar3131.
38. Tiyaboonchai,A., Cardenas-Diaz,F.L., Ying,L., Maguire,J.A., Sim,X., Jobaliya,C., Gagne,A.L., Kishore,S., Stanescu,D.E., Hughes,N., *et al.* (2017) GATA6 Plays an Important Role in the Induction of Human Definitive Endoderm, Development of the Pancreas, and Functionality of Pancreatic β Cells. *Stem cell reports*, **8**, 589–604.
39. Ng,E.S., Azzola,L., Sourris,K., Robb,L., Stanley,E.G. and Elefanty,A.G. (2005) The primitive streak gene Mixl1 is required for efficient haematopoiesis and BMP4-induced ventral mesoderm patterning in differentiating ES cells. *Development*, **132**.
40. Jagtap,S., Meganathan,K., Gaspar,J., Wagh,V., Winkler,J., Hescheler,J. and Sachinidis,A. (2011) Cytosine arabinoside induces ectoderm and inhibits mesoderm expression in human embryonic stem cells during multilineage differentiation. *Br. J. Pharmacol.*, **162**, 1743–56.
41. Ng,V.Y., Ang,S.N., Chan,J.X. and Choo,A.B.H. (2010) Characterization of Epithelial Cell Adhesion Molecule as a Surface Marker on Undifferentiated Human Embryonic Stem Cells. *Stem Cells*, **28**, 29–35.
42. Thomas,T., Nowka,K., Lan,L. and Derwahl,M. (2006) Expression of Endoderm Stem Cell Markers: Evidence for the Presence of Adult Stem Cells in Human Thyroid Glands. *Thyroid*, **16**, 537–544.
43. Bron,C. and Kerbosch,J. (1973) Algorithm 457: finding all cliques of an undirected graph. *Commun. ACM*, **16**, 575–577.
44. Davidson,K.C., Adams,A.M., Goodson,J.M., McDonald,C.E., Potter,J.C., Berndt,J.D., Biechele,T.L., Taylor,R.J. and Moon,R.T. (2012) Wnt/β-catenin signaling promotes differentiation, not self-renewal, of human embryonic stem cells and is repressed by Oct4. *Proc. Natl. Acad. Sci. U. S. A.*, **109**, 4485–90.
45. Kolodziejczyk,A.A., Kim,J.K., Tsang,J.C.H., Ilicic,T., Henriksson,J., Natarajan,K.N., Tuck,A.C., Gao,X., Bühler,M., Liu,P., *et al.* (2015) Single Cell RNA-Sequencing of Pluripotent States Unlocks Modular Transcriptional Variation. *Cell Stem Cell*, **17**, 471–485.
46. Pollen,A.A., Nowakowski,T.J., Shuga,J., Wang,X., Leyrat,A.A., Lui,J.H., Li,N., Szpankowski,L., Fowler,B., Chen,P., *et al.* (2014) Low-coverage single-cell mRNA sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex. *Nat. Biotechnol.*, **32**, 1053–8.
47. Usoskin,D., Furlan,A., Islam,S., Abdo,H., Lönnerberg,P., Lou,D., Hjerling-Leffler,J.,

Haeggström,J., Kharchenko,O., Kharchenko,P. V, *et al.* (2014) Unbiased classification of sensory neuron types by large-scale single-cell RNA sequencing. *Nat. Neurosci.*, **18**, 145–153.

48. Villani,A.-C., Satija,R., Reynolds,G., Sarkizova,S., Shekhar,K., Fletcher,J., Griesbeck,M., Butler,A., Zheng,S., Lazo,S., *et al.* (2017) Single-cell RNA-seq reveals new types of human blood dendritic cells, monocytes, and progenitors. *Science*, **356**, eaah4573.
49. Yeung,K.Y. and Ruzzo,W.L. (2001) Details of the Adjusted Rand index and Clustering algorithms Supplement to the paper " An empirical study on Principal Component Analysis for clustering gene expression data " (to appear in Bioinformatics).
50. Chu,L.-F., Leng,N., Zhang,J., Hou,Z., Mamott,D., Vereide,D.T., Choi,J., Kendziorski,C., Stewart,R. and Thomson,J.A. (2016) Single-cell RNA-seq reveals novel regulators of human embryonic stem cell differentiation to definitive endoderm. *Genome Biol.*, **17**, 173.
51. Moignard,V., Macaulay,I.C., Swiers,G., Buettner,F., Schütte,J., Calero-Nieto,F.J., Kinston,S., Joshi,A., Hannah,R., Theis,F.J., *et al.* (2013) Characterization of transcriptional networks in blood stem and progenitor cells using high-throughput single-cell gene expression analysis. *Nat. Cell Biol.*, **15**, 363–72.
52. Angerer,P., Simon,L., Tritschler,S., Wolf,F.A., Fischer,D. and Theis,F.J. (2017) Single cells make big data: New challenges and opportunities in transcriptomics. *Curr. Opin. Syst. Biol.*, **4**, 85–91.
53. Sathyamurthy,A., Johnson,K.R., Matson,K.J.E., Dobrott,C.I., Li,L., Ryba,A.R., Bergman,T.B., Kelly,M.C., Kelley,M.W. and Levine,A.J. (2018) Massively Parallel Single Nucleus Transcriptional Profiling Defines Spinal Cord Neurons and Their Activity during Behavior. *Cell Rep.*, **22**, 2216–2225.
54. Levine,E. and Domany,E. (2001) Resampling Method for Unsupervised Estimation of Cluster Validity. *Neural Comput.*, **13**, 2573–2593.
55. Papili Gao,N., Ud-Dean,S.M.M., Gandrillon,O. and Gunawan,R. (2017) SINCERITIES: Inferring gene regulatory networks from time-stamped single cell transcriptional expression profiles. *Bioinformatics*, 10.1093/bioinformatics/btx575.
56. Smoot,M.E., Ono,K., Ruscheinski,J., Wang,P.-L. and Ideker,T. (2011) Cytoscape 2.8: new features for data integration and network visualization. *Bioinformatics*, **27**, 431–2.

TABLES AND FIGURES

Table 1. Comparison of single-cell clustering algorithms

Dataset	MONOCLE 2	SC3	SIMLR	CALISTA
Kolodziejczyk et al. (704 cells, 3 clusters)	0.97	1	1	0.96
Pollen et al. (249 cells, 11 clusters)	0.86	0.95	0.94	0.89
Usoskin et al. (622 cells, 4 clusters)	0.57	0.88	0.68	0.91
Villani et al. (751 cells, 6 clusters)	0.58	0.88	0.37	0.86

Table 2. Comparison of lineage progression inference

	Lineage Inference			
	MONOCLE 2	***DPT***	***SCUBA***	***CALISTA***
Inferrable bifurcation event/-s	Multiple	Single	Multiple	Multiple
Required *a priori* information for the method	None	Linear/Branching	Time Points	None
Bargaje (1896 cells, 1 bifurcation event)	✓	✗	✓	✓
Chu (728 cells, no bifurcation event)	✗	✗	✗	✓
Moignard (597 cells, 2 bifurcation events)	✓ Partial	✓ Partial	–	✓
Treutlein (405 cells, 1 bifurcation event)	✓	✓	✓	✓
✓ = trajectory inferred correctly				
✗ = trajectory inferred incorrectly				
✓partial = trajectory inferred only partially correct				

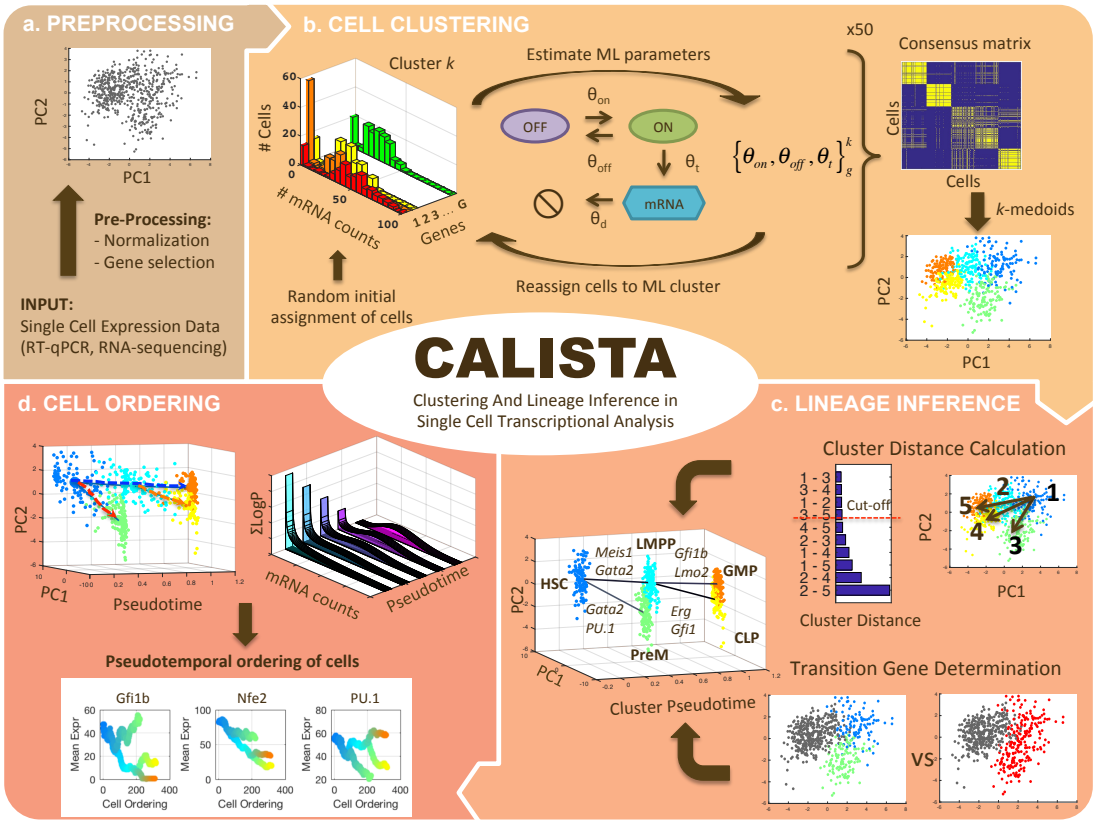

Figure 1. CALISTA workflow. (a) Preprocessing of single-cell expression input data. (b) Single-cell clustering in CALISTA combines maximum likelihood and consensus clustering approaches. In the maximum likelihood step, CALISTA relies on the two-state model of the

gene transcription process in combination with a model of random dropouts to describe the distribution of single-cell gene expression. The maximum likelihood clustering is implemented multiple times, each starting from random cell assignment into clusters, using a greedy algorithm to generate a consensus matrix. The final step implements *k*-medoids clustering algorithm using the consensus matrix. (c) CALISTA uses cluster distances - a measure of dissimilarity in the gene expression distribution between any two clusters – to reconstruct the lineage progression graph. The state transition edges are added in increasing magnitude of cluster distances. In addition, CALISTA provides transition genes between any two connected clusters in the lineage graph based on the gene-wise likelihood differences of having the cells separately and having them in one cluster. (d) For pseudotemporal ordering of the cells, CALISTA reassigns each cell to a transition edge and computes its pseudotime by maximizing the cell likelihood. CALISTA uses a linear interpolation for evaluating the cell likelihood between two connected clusters. Given a developmental path in the lineage progression, CALISTA generates moving averaged gene expression trajectories using the pseudotemporally ordered cells in the path.

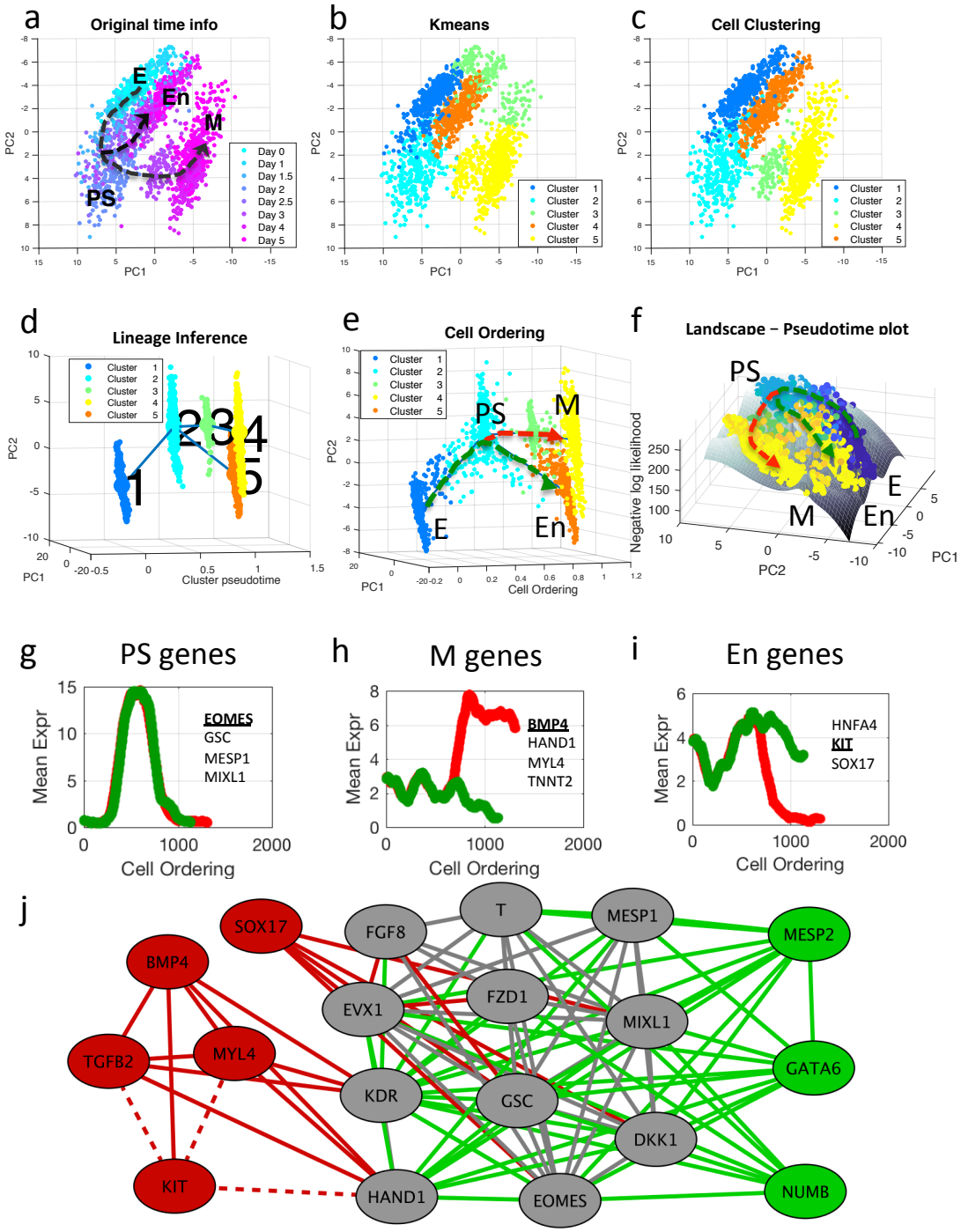

Figure 2. Analysis of single-cell transcriptional profiles during iPSC differentiation into cardiomyocyte. The single-cell gene expression dataset was taken from the study of Bargaje et al. (26) (a-c) PCA plots of single-cell dataset along the first two principal components. Each dot represents a cell, where the colors indicate (a) the capture time info, (b) *k*-means single-cell clusters, and (c) CALISTA single-cell clusters. (d) Lineage progression graph reconstructed by CALISTA. The graph was generated by adding transition edges in increasing order of cluster distances until each cluster is connected by an edge. Following this procedure, an edge connecting cluster 1 and 5 was originally added in the graph. We

manually removed this edge as the edge bypassed intermediate capture times. (e) Pseudotemporal ordering of cells. Cells were first assigned to the state transition edges in the lineage progression, and then pseudotemporally ordered along two developmental paths – mesodermal (M) path (red) and endodermal (En) path (green). (f) Cell likelihood landscape. The landscape shows the surface of the negative log-likelihood of the cells. A higher value thus indicates a state of increased gene expression uncertainty. (g-i) Moving window average of gene expression along M and En developmental paths for representative (g) PS, (h) M, and (i) En genes (underlined and boldfaced). (j) Gene modules of M (red and grey) and En paths (green and grey) (drawn using Cytoscape (56)). The grey nodes and edges are common between the M and En modules. Solid (dashed) lines indicate positive (negative) gene correlations.

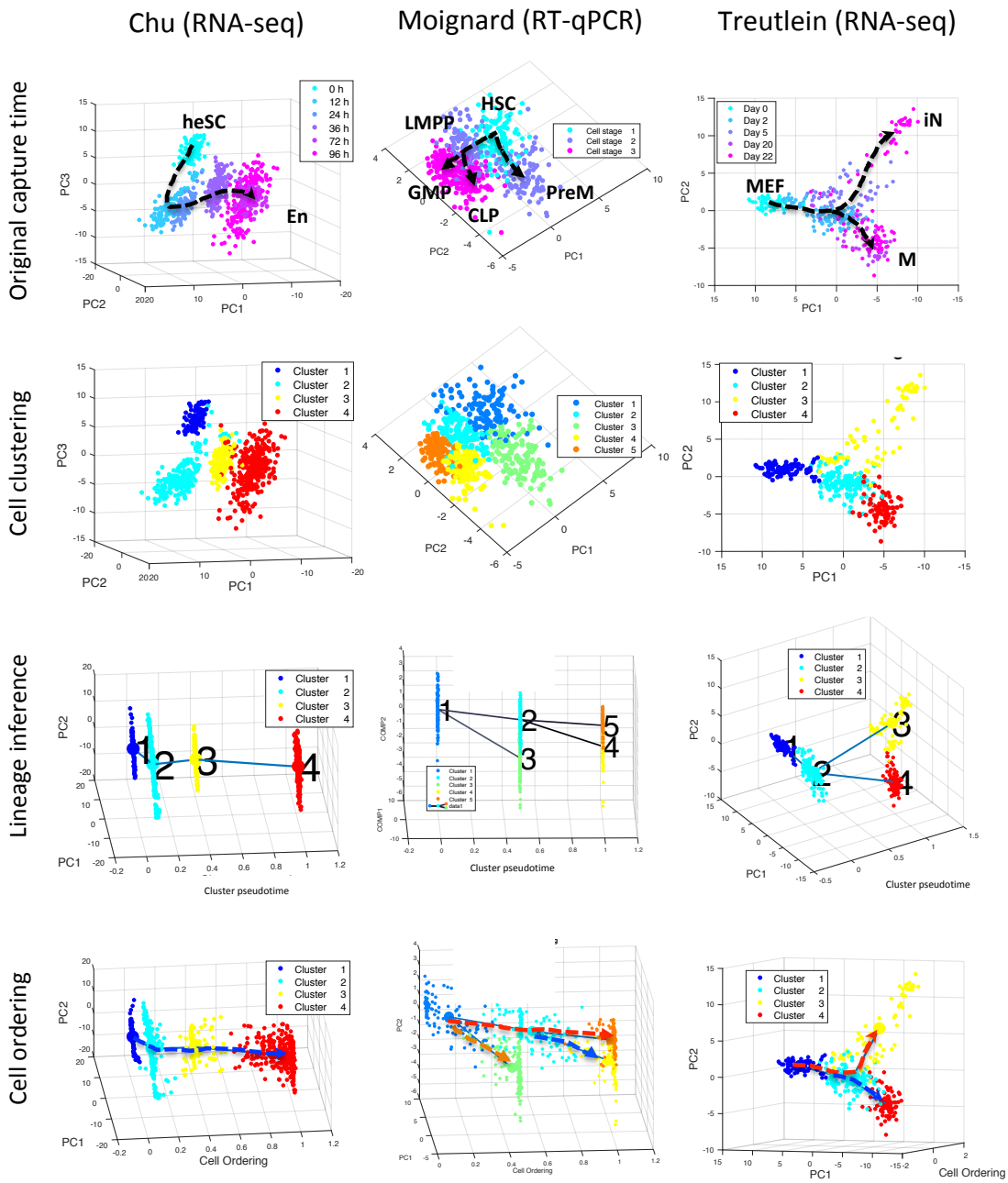

Figure 3. CALISTA analysis of single-cell transcriptional profiles. (Left) Human embryonic stem cell differentiation into endodermal cells in Chu et al. study (50). (Middle) Hematopoietic stem cell differentiation in Moignard et al. study (51). (Right) Mouse embryonic fibroblast differentiation into neurons in Treutlein et al. study (11).

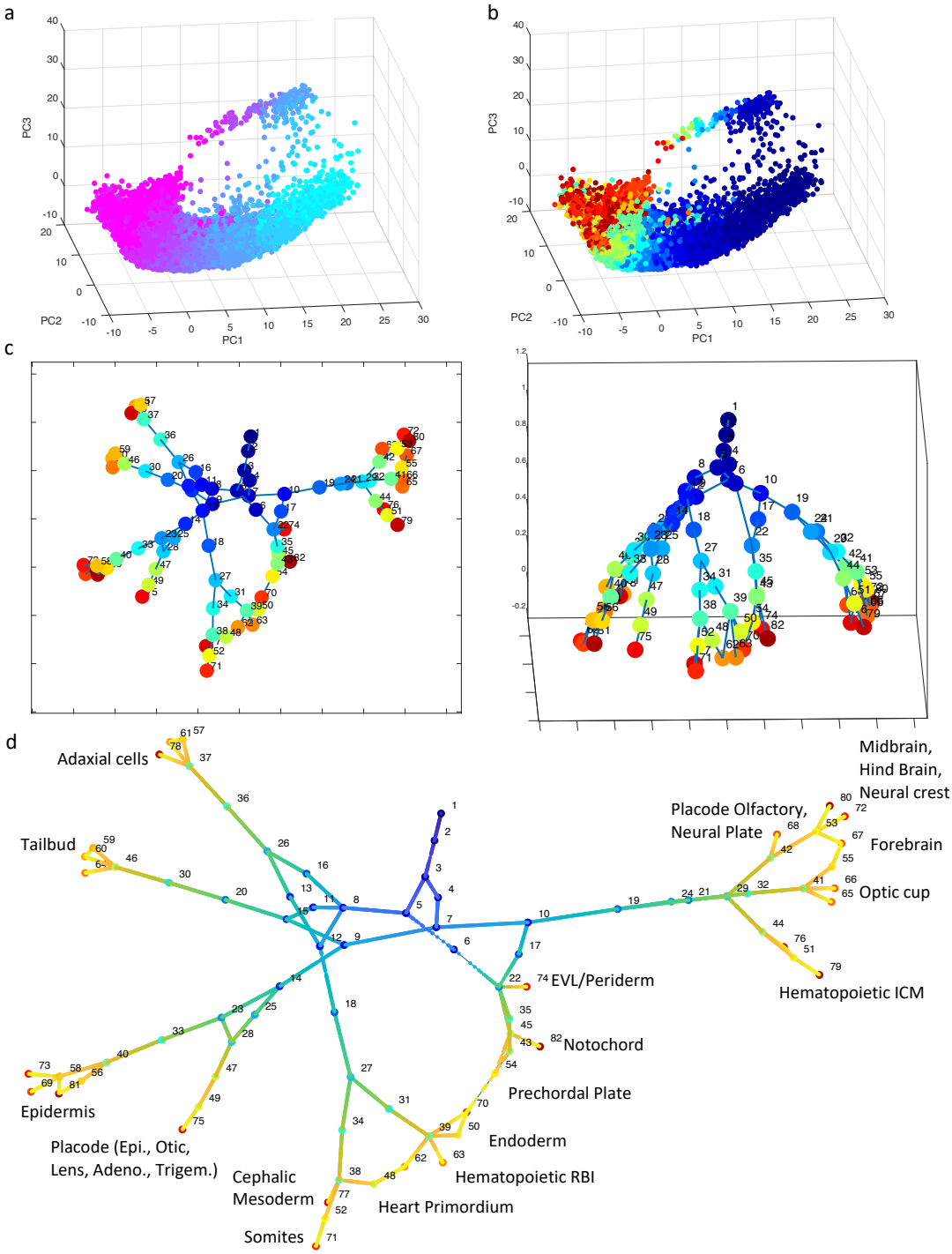

Figure 4. CALISTA analysis of developmental trajectories during zebrafish embryogenesis. The single-cell Drop-seq expression dataset was taken from the study of Farrell et al. (37) (a-

b) PCA plots of single-cell dataset along the first two principal component. Each symbol represents a cell, while the colors represent (a) the capture time info (from light blue = stage 1 to magenta = stage 12), (b) CALISTA single-cell clusters, (c) reconstruction of lineage progression (see Additional file 5), and (d) pseudotemporal ordering of cells (from blue = 0 to yellow = 1). Cells were first assigned to the state transition edges in the lineage progression, and pseudotemporally ordered along developmental paths.