bioRxiv preprint first posted online Oct. 25, 2017; doi: http://dx.doi.org/10.1101/208819. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.

October 25, 2017

Graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells

F. Alexander Wolf[1], Fiona Hamey[2], Mireya Plass[3], Jordi Solana[3], Joakim S. Dahlin[2,4], Berthold Göttgens[2], Nikolaus Rajewsky[3], Lukas Simon[1] & Fabian J. Theis[1,5,†]

1 Helmholtz Zentrum München – German Research Center for Environmental Health, Institute of Computational Biology, Neuherberg, Munich, Germany.

2 Department of Haematology and Wellcome Trust-Medical Research Council Cambridge Stem Cell Institute, University of Cambridge, Cambridge, United Kingdom.

3 Berlin Institute for Medical Systems Biology, Max-Delbrück Center for Molecular Medicine, Germany.

4 Department of Medicine, Karolinska Institutet and Karolinska University Hospital, Stockholm, Sweden.

5 Department of Mathematics, Technische Universität München, Munich, Germany.

† fabian.theis@helmholtz-muenchen.de

Abstract

Single-cell RNA-seq allows quantification of biological heterogeneity across both discrete cell types and continuous cell differentiation transitions. We present approximate graph abstraction (AGA), an algorithm that reconciles the computational analysis strategies of clustering and trajectory inference by explaining cell-to-cell variation both in terms of discrete and continuous latent variables (https://github.com/theislab/graph_abstraction). This enables to generate cellular maps of differentiation manifolds with complex topologies — efficiently and robustly across different datasets.

Extended Abstract

Approximate graph abstraction quantifies the connectivity of partitions of a neighborhood graph of single cells, thereby generating a much simpler abstracted graph whose nodes label the partitions. Together with a random walk-based distance measure, this generates a topology preserving map of single cells — a partial coordinatization of data useful for exploring and explaining its variation. We use the abstracted graph to assess which subsets of data are better explained by discrete clusters than by a continuous variable, to trace gene expression changes along aggregated single-cell paths through data and to infer abstracted trees that best explain the global topology of data. We demonstrate the power of the method by reconstructing differentiation processes with high numbers of branchings from single-cell gene expression datasets and by identifying biological trajectories from single-cell imaging data using a deep-learning based distance metric. Along with the method, we introduce measures for the connectivity of graph partitions, generalize random-walk based distance measures to disconnected graphs and introduce a path-based measure for topological similarity between graphs. Graph abstraction is computationally efficient and provides speedups of at least 30 times when compared to algorithms for the inference of lineage trees.

Introduction

Single-cell RNA-Seq offers unparalleled opportunities for comprehensive molecular profiling of thousands of individual cells, with expected major impacts across a broad range of biomedical research.

bioRxiv preprint first posted online Oct. 25, 2017; doi: http://dx.doi.org/10.1101/208819. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.

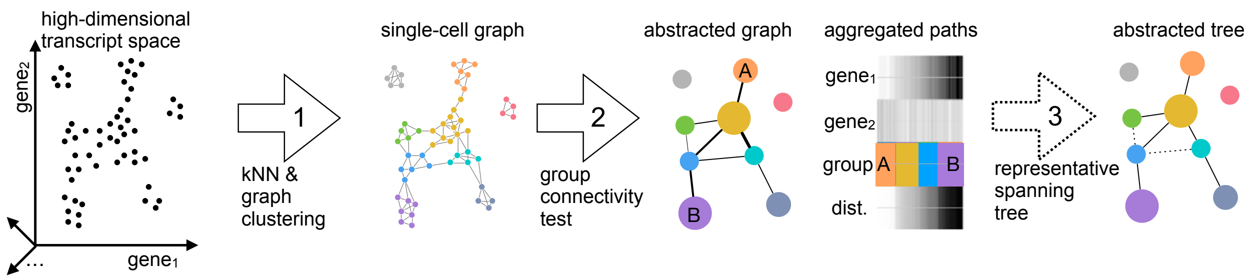


Figure 1 | Approximate graph abstraction generates a topology preserving map of single cells. Sketch of high-dimensional gene expression data and a partitioned k-nearest-neighbor graph $\mathcal{G}$ of data points, whose nodes represent single cells and whose partitions represent groups of similar cells. From this, we generate a much simpler abstracted graph $\mathcal{G}^*$ whose nodes correspond to partitions of $\mathcal{G}$ and whose edges represent statistically estimated connectivity between partitions. Hence, the abstracted graph $\mathcal{G}^*$ represents the topology of $\mathcal{G}$ at the coarse resolution of the partitioning. The statistical model used to generate the abstracted graph $\mathcal{G}^*$ allows to identify high-confidence paths through groups — those along thick edges. Using a random-walk based distance measure d for potentially disconnected graphs, we order cells within each group according to their distance from a root cell. A path that passes through a sequence of nodes in $\mathcal{G}^*$ then aggregates all single-cell paths in $\mathcal{G}$ that pass through the corresponding groups of cells. This allows to trace gene changes along complex trajectories at single-cell resolution. Hence, graph abstraction provides a coordinate system $(\mathcal{G}^*, d)$ — a topology preserving map — of cells. The abstraction of many biological processes, such as development, is thought to be tree-like. Hence, we provide an algorithm for identifying the abstracted tree $\mathcal{T}^*$ (solid edges) that best approximates the topology of $\mathcal{G}$.

The resulting datasets are often discussed using the term transcriptional landscape. However, the algorithmic analysis of cellular heterogeneity and patterns across such landscapes still faces fundamental challenges, for instance, in how cell-to-cell variation is explained. Current computational approaches attempt to achieve this in only one of two ways [1]. Clustering assumes that data is composed of biologically distinct groups of cells such as discrete cell types and labels these with a discrete variable — the cluster index. Inferring pseudotemporal orderings or trajectories of cells [2, 3], by contrast, assumes that data lie on a connected manifold [4] and labels cells with a continuous variable — the distance along the manifold. While the former approach is the basis for most unsupervised analyses of single-cell data, the latter enables a better interpretation of continuous phenotypes and processes such as development, dose response and disease progression. Here, we unify both viewpoints in a method to which we refer as approximate graph abstraction (AGA) [5].

A central example of dissecting heterogeneity in single-cell experiments concerns data that originate from complex cell differentiation processes. However, analyzing such data using pseudotemporal ordering [2, 6–10] faces the problem that biological processes are usually incompletely sampled. As a consequence, experimental data does not conform with a connected manifold and the modeling of data as a continuous tree structure, which is the basis for existing algorithms, has little meaning (Supplemental Note 1). Even the clustering-based algorithms for the inference of tree-like processes [11–13] make the generally invalid assumption that clusters conform with a connected tree-like topology. Moreover, they face the problem of relying on simple, fixed distance measures, which quantify biological similarity of cells only at a local scale [2, 3, 8] and are fraught with problems when used for larger-scale objects like clusters. Efforts for addressing the resulting high non-robustness of tree-fitting to distances between clusters [11] by sampling [12, 13] have only had limited success (Supplemental Note 1).

Approximate graph abstraction resolves these fundamental problems by generating a map of cells that preserves both continuous and disconnected structure in data, that is, the topology of data. The data-driven formulation of graph abstraction enables to robustly reconstruct branching gene expression changes across different datasets and, for the first time, to reconstruct the lineage tree of a whole adult animal [14].

bioRxiv preprint first posted online Oct. 25, 2017; doi: http://dx.doi.org/10.1101/208819. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.

Results

Approximate graph abstraction explains both discrete and continuous cell-to-cell variation.

We represent single-cell gene expression data using the single-cell graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, whose nodes $\mathcal{N}$ correspond to gene expression measurements of cells and whose edges $\mathcal{E}$ indicate neighborhood between cells (Figure 1) [e.g. 3, 15, 16]. The complexity of $\mathcal{G}$ and noise-related spurious edges make it both hard to trace a putative biological process from progenitor cells to different fates and to decide whether groups of cells are connected or disconnected. Moreover, tracing isolated paths of single cells to make statements about a biological process comes with too little statistical power to achieve an acceptable confidence level. Gaining statistical power by averaging over distributions of single-cell paths is hampered by the difficulty of fitting realistic models for the distribution of these paths (Supplemental Note 2).

We address these problems by, first, developing a statistical model for the connectivity of groups of cells (Supplemental Note 3), which we determine by graph-partitioning [15, 17] (Supplemental Note 4). This allows us to generate a much simpler abstracted graph $\mathcal{G}^*$ (Figure 1) whose nodes correspond to groups and whose edge weights quantify the connectivity between groups. Hence, while $\mathcal{G}$ represents the connectivity structure of data at single-cell resolution, the abstracted graph $\mathcal{G}^*$ represents the connectivity structure of data at the coarse resolution of the partitioning. Following paths along nodes in $\mathcal{G}^*$ means following an ensemble of single-cell paths that pass through the corresponding groups in $\mathcal{G}$. By averaging over such an ensemble of single-cell paths it becomes possible to trace a putative biological process from a progenitor to fates in a way that is robust to spurious edges, provides statistical power and is consistent with basic assumptions on a biological trajectory of cells (Supplemental Note 2).

To trace gene dynamics at single-cell resolution, we extended existing scale-free random-walk based distance measures d (Supplemental Note 2, Reference [8]) to the realistic case that accounts for disconnected graphs. By following high-confidence paths in the abstracted graph $\mathcal{G}^*$ and ordering cells within each group in the path according to their distance d from a progenitor cell, we trace gene dynamics at single-cell resolution (Figure 1). Hence, graph abstraction covers both aspects of clustering and pseudotemporal ordering by providing a topology preserving map of cells: a coordinate system $(\mathcal{G}^*, d)$ that allows to explore and explain variation in data while being faithful to its connectivity structure (Supplemental Note 5). Graph abstraction can be viewed as an easily-interpretable, scalable and robust way of performing topological data analysis for potentially disconnected data (Supplemental Note 1).

Graph abstraction enables the consistent reconstruction of gene expression dynamics across datasets.

In myeloid hematopoiesis, progenitor cells differentiate into erythrocytes, megakaryoctyes, monocytes, neutrophils and other blood cell types. We consider a computational model of this system (Supplemental Note 6) and two experimental datasets based on the MARS-seq [18] and Smart-seq2 protocol [19], respectively. Neighborhood relations among cells and thus the single-cell graph $\mathcal{G}$ are generated using established preprocessing steps with default parameters [16, 20–22] and $\mathcal{G}$ is visualized using the Fruchterman-Reingold [FR, 23, 24] algorithm, which conserves continuous structure in data better than tSNE (Supplemental Figure 6, Reference [16]). While the edges of the single-cell graph $\mathcal{G}$ are too many to be meaningfully shown, we can visualize the abstracted graph $\mathcal{G}^*$ using a simple tree-based graph drawing layout (Figure 2a, c).

Even though data from different experimental protocols and simulations are involved, the abstracted graphs (Figure 2a, c and Supplemental Figures 6 and 7) between all results are consistent using default parameters. Moreover, we find consistent continuous gene expression changes when tracing paths of highest-confidence in $\mathcal{G}^*$ to the erythrocyte, monocyte and neutrophil fate for the marker genes *Gata2*, *Gata1*, *Hba-a2*, *Elane*, *Gfi1*, *Irf8* and *Csf1r*. All agree well with the known behavior of these genes during commitment towards the different blood fates: The expression of *Gata2* slowly decreases along all paths. Markers for the erythroid fate, *Gata1* and *Hba-a2*, become expressed at the

bioRxiv preprint first posted online Oct. 25, 2017; doi: http://dx.doi.org/10.1101/208819. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.

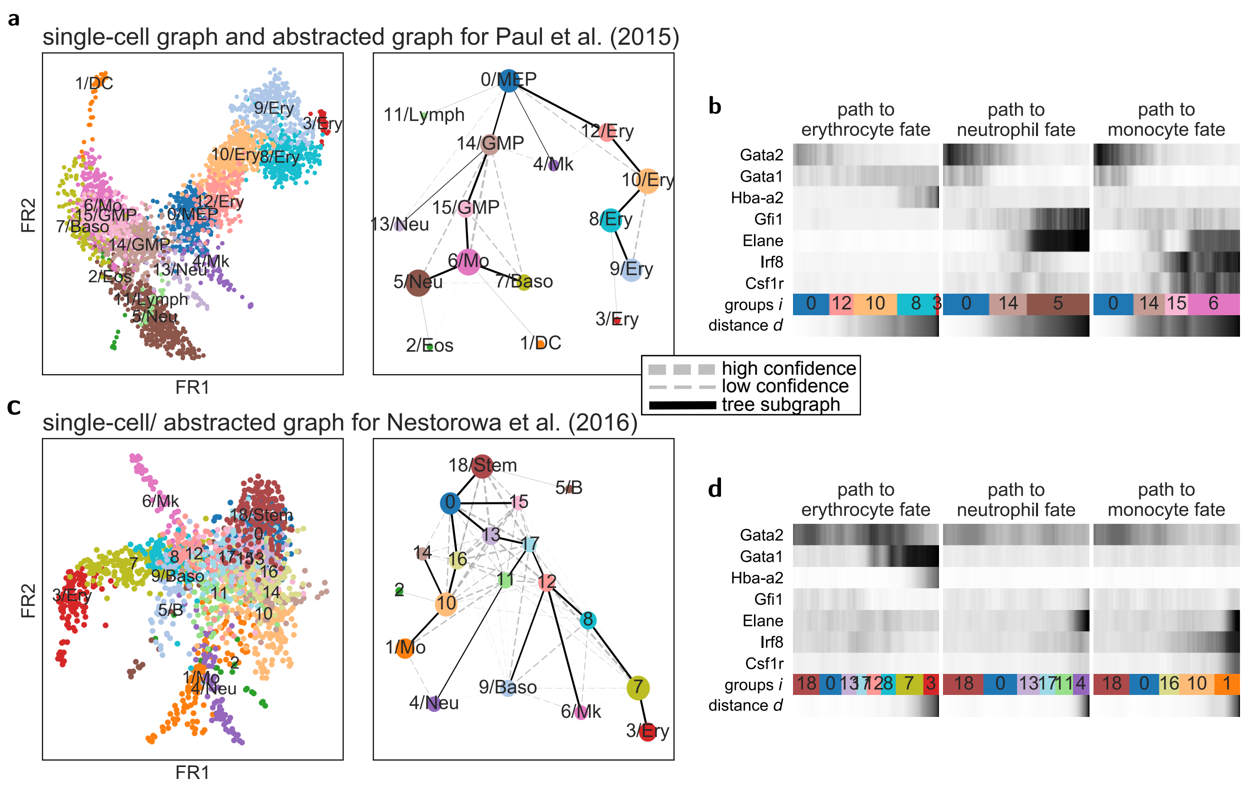


Figure 2 | Graph abstraction enables consistent reconstruction of gene expression dynamics across datasets. **a,** Fruchterman-Reingold (FR) visualization of single-graph for myeloid differentiation for data of Paul *et al.* [18] and the abstracted graph in which edge width is proportional to the confidence in the connectedness of cell groups. The tree-like subgraph that best explains the topology of the data is highlighted with solid edges. However, there are several connections of high confidence that do not confirm with a tree topology. The cell groups within the single-cell graph and the nodes of the abstracted graph are labelled with the cell type of highest overlap identified by Paul *et al.* [18]. The abbreviations are MEP for myeloid-erythrocyte progenitor, Mk for megakaryocytes, Lymph for lymphocytes, Ery for erythrocytes, GMP for granulocyte/macrophage progenitors, Mo for monocytes, Baso for basophils, Neu for neturophils, DC for dendritic cells and Eos for eosinophils. **b,** Continuous gene expression changes along the highest-confidence paths — which include dashed edges — to erythrocytes, neutrophils and monocytes in the abstracted graph. Within each group, cells are ordered according to a random-walk based distance from the earliest progenitor cell in the 0/MEP group. **c,** The data of Nestorowa *et al.* [19] displays a much higher level of non-tree like geometry, more similar to a cloud, due to a higher content of heterogeneous stem cells. **d,** Gene expression changes along the highest-confidnce paths to erythrocytes, neutrophils and monocytes are consistent with those observed in panel c. However, as the data has been measured earlier in hematopoiesis, changes in marker genes occur later during progression along paths.

late stages only along the paths to erythrocytes. *Gfi1* and *Elane* are only activated along the paths to Neutrophils and *Irf8* and *Csf1r* are only activated along the paths to monocytes (Figure 2b, d and Supplemental Figures 6 and 7). Remarkably, the connectivity structure shown in Figure 2c is not clear enough to decide whether there is a shared megakaryocyte-erythrocyte-basophil progenitor, as has previously been suggested to exist in human [25], or whether neutrophils, basophils and monocytes originate separately from the erythroid lineage (see also Supplemental Figure 7). Below, we discuss that algorithms for the inference of lineage trees only allow to obtain non-robust trees for the data of Paul *et al.* [18] (Supplemental Figure 3) and are not able to produce a meaningful result for the data of Nestorowa *et al.* [19] (Supplemental Figure 4) because the latter contains a high fraction of heterogeneous stem cells and strongly deviates from a tree topology. Such behavior during differentiation has been observed before and termed “cloud of cells” [26].

Graph abstraction enables inferring the lineage tree of a whole adult animal.

The complicated global topology of $\mathcal{G}$ represents the information about the continuity of biological

bioRxiv preprint first posted online Oct. 25, 2017; doi: http://dx.doi.org/10.1101/208819. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.

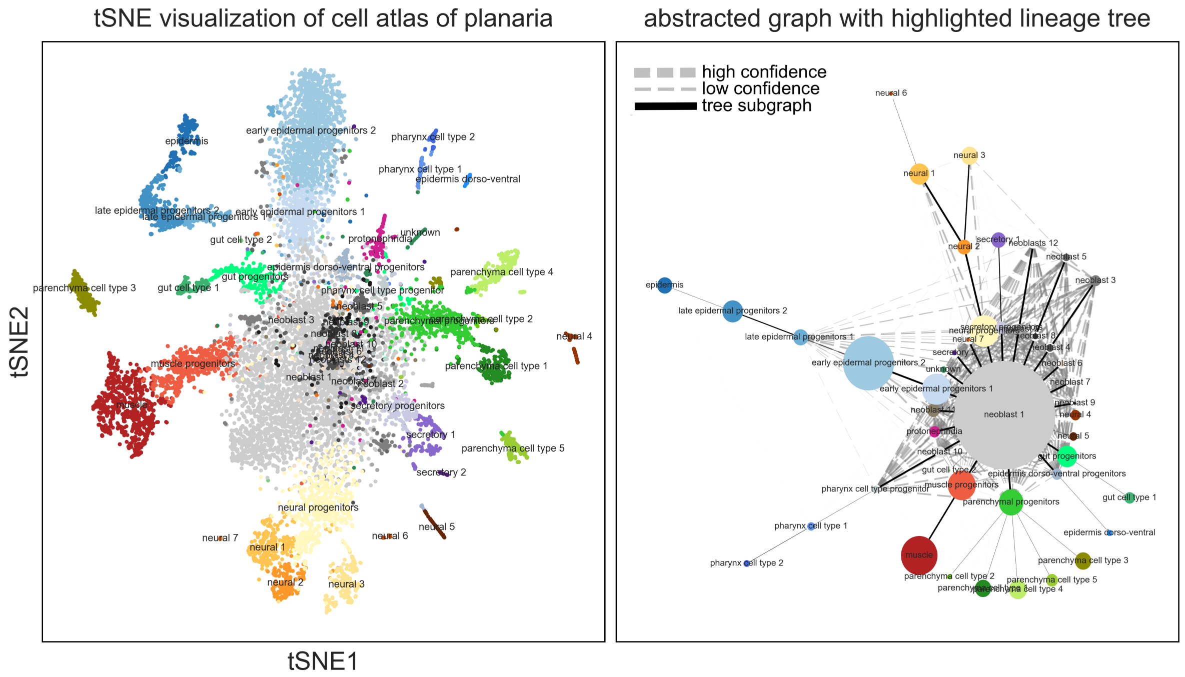


Figure 3 | Abstracted graph and lineage tree of a whole adult animal: planaria [14]. tSNE visualization of cell groups and reconstruction of the abstracted graph $\mathcal{G}^*$ and a likely candidate for a lineage tree $\mathcal{T}^*$. In the tSNE, many cell groups appear disconnected. The abstracted graph, by contrast, recovers all continuous differentiation paths.

processes measured in single-cell data. The abstractions of many cellular processes though, such as differentiation in several organisms and tissues, are known to be well-described by simple tree-like topologies [2, 6, 13, 27]. In the framework of graph abstraction, we would hence like to identify the tree-like subgraph $\mathcal{T}^*$ in $\mathcal{G}^*$ that best conforms with the global topology of $\mathcal{G}$, that is, contains most of its paths at a coarse-grained resolution. We address this by finding the tree that maximizes overall connectivity and thus continuity between any two groups of cells. While this is solved by the minimum spanning tree applied to $\mathcal{G}^*$ when weighted by inverse connectivity, for realistic cases, many almost degenerate solutions of maximal connectivity exist. To address this, we propose to approximate the global topology by a set of points in $\mathcal{G}$ that maximize d, that is, that "stick out" most. Extending Prim's minimum spanning tree algorithm, we iteratively match these extremal points to leaf nodes of a growing $\mathcal{T}^*$ while maximizing connectivity (Supplemental Note 8).

Using this algorithm, graph abstraction enables the reconstruction of the lineage tree of a whole adult animal, planaria, using single-cell gene expression data from 12,252 cells (Figure 3). Due to the unique regenerative capabilities of planaria, a single-cell snapshot of an adult planaria captures all transitional cells and we can — as data is sampled densely enough — observe all lineage relations of its cell atlas with only few weakly connected cell types. The tree inferred with graph abstraction predicts the existence of at least 20 independent cell lineages in planaria, for example, the major tissues such as neurons, muscle, parenchyma and gut. Among the 20 lineages, only the epidermic lineage has been previously characterized at the transcriptomic level. The details of this and the insight that the degree of nodes in the abstracted graph $\mathcal{G}^*$ characterize potency of cell types better than previous measures [13] are discussed in a separate publication [14]. It is important to realize that if we had followed a traditional analysis based on tSNE and clustering, we would have concluded to clean the data by removing the 14 seemingly disconnected clusters shown in Figure 3.

Graph abstraction is more robust than algorithms for the reconstruction of branching lineages.

To assess how robustly graph and tree-inference algorithms recover a given topology, we developed

bioRxiv preprint first posted online Oct. 25, 2017; doi: http://dx.doi.org/10.1101/208819. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.

a measure for comparing the topologies of two graphs by comparing the sets of possible paths on them (Supplemental Note 7, Supplemental Figure 8). Sampling widely varying parameters, which leads to widely varying clusterings, we find that the inferred abstraction of topology of data within AGA is much more robust than the underlying graph clustering algorithm (Supplemental Figure 9). This is reassuring as graph clustering alone is, as any clustering method, an ill-posed problem in the sense that many highly degenerate quasi-optimal clusterings exist and some knowledge about the scale of clusters is required.

Several algorithms [6, 11–13] have been proposed for reconstructing lineage trees (Supplemental Note 1). The main caveat of these algorithms is that they, unlike graph abstraction, try to explain any variation in the data with a tree-like topology, which they model with models with tree-like geometry. In particular, any disconnected distribution of clusters is interpreted as originating from a tree. This produces qualitatively erroneous results (Supplementary Figure 1) already for the minimal example of Supplementary Figure 6 and only works well for data that clearly conforms with a tree-like manifold (Supplementary Figure 2). To establish a fair comparison on real data with the most recent method, Monocle 2, we reinvestigated the main example of Qiu *et al.* [6] for a complex differentiation tree. This example is based on the data of Paul *et al.* [18] (Figure 2), but with a cluster of lymphoid cells removed. While graph abstraction yields consistent results independent of the presence of this cluster, the prediction of Monocle 2 becomes qualitatively wrong if the cluster is present (Supplementary Figure 3). The example illustrates the general point that data always consists of dense and sparse — connected and disconnected — regions, some tree-like, some with more complex topology. Simplifying the topology by manually removing clusters that appear disconnected in tSNE but may in fact be connected (Supplemental Figure 5) would, for instance, have precluded us from inferring the lineage tree of planaria (Figure 3).

Graph abstraction is a powerful and versatile tool for analyzing large single-cell data.

We demonstrate the scalability of graph abstraction by computing the abstracted graph for more than 68 000 peripheral blood mononuclear cells (PBMCs) [21], which took around 2 minutes (Supplemental Figure 10). Comparing runtimes with Monocle 2 and stemID 2 [13], we find speedups of 38 and 309 times, respectively (Supplemental Note 1). Moreover, the PBMC dataset serves as a negative control for the predictions of graph abstraction as it mostly lacks cells in the transitioning stages between different cell types. Hence, only crude motifs of the lineage tree of PBMCs are reconstructed. CD34+ progenitor cells are only weakly connected to the rest of the data. CD14+ Monocytes, CD56+ NK cells, CD19 B cells and CD45RO+ Memory cells are correctly placed at leaves of the abstracted graph and the connectivity of CD4+ and CD8+ cells is reflected in thick edges in the abstracted graph (Supplemental Figure 10). Consistent results are observed for a further dataset obtained using different chemistry (Supplemental Figure 10).

Discussion

With challenges such as the Human Cell Atlas [28], there is a strongly increasing need for aggregating and consistently modeling biological questions across different experimental setups. After recent advances in the study of simple biological processes that involve a single branching [7, 8], graph abstraction provides a similarly robust framework for arbitrarily complex trajectories.

In closing, we note that graph abstraction can be extended to analyze single-cell imaging data when applied on the basis of a deep-learning based distance metric. Eulenberg *et al.* [29] showed that a deep learning model can generate a feature space in which distances reflect the continuous progression of cell cycle and a disease. Using this, graph abstraction correctly identifies the biological trajectory through the interphases of cell cycle while ignoring a cluster of damaged and dead cells (Supplemental Note 10).

bioRxiv preprint first posted online Oct. 25, 2017; doi: http://dx.doi.org/10.1101/208819. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.

Code availability

Approximate graph abstraction is available within Scanpy `https://github.com/theislab/scanpy`. The analyses and results of the present paper are available from `https://github.com/theislab/graph_abstraction`.

Data availability

Data is available from `https://github.com/theislab/graph_abstraction`.

Acknowledgements

We thank N. Yosef for stimulating discussions and important comments, S. Tritschler for valuable feedback when testing the code and M. Luecken for comments on graph partitioning algorithms. F.A.W. acknowledges support by the Helmholtz Postdoc Programme, Initiative and Networking Fund of the Helmholtz Association. J.S.D. is supported by a grant from the Swedish Research Council.

bioRxiv preprint first posted online Oct. 25, 2017; doi: http://dx.doi.org/10.1101/208819. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.

Supplemental Notes

Contents

Supplemental Note 1: Comparing graph abstraction with other algorithms

Establishing fair comparisons with other algorithms is difficult mainly for two reasons. (i) Comparisons on real data are problematic as a quantitative undebatable ground truth is hard to obtain. (ii) It is very easy to "make algorithms fail", for example, by choosing an unsuitable preprocessing or pathologic parameters. Hence, after a comparison of concepts (Supplemental Note 1.1), we restrict ourselves to addressing fundamental problems and qualitatively wrong predictions of other algorithms for simple minimal examples with known ground truth (Supplemental Note 1.2), well-known

bioRxiv preprint first posted online Oct. 25, 2017; doi: http://dx.doi.org/10.1101/208819. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.

real data, where we were able to reproduce published testing conditions [6] (Supplemental Note 1.3 and 1.4) and a comparison of runtimes (Supplemental Note 1.5).

Supplemental Note 1.1: Comparison of concepts

Monocle 2 [6] uses "reversed graph embedding" [31], which aims to fit a geometrical model for a graph to projections of the data to a low-dimensional latent space. Even though, in principle, any model could be used for that, in practice, only tree-like models are computationally tractable. Hence, Monocle 2 tries to force data into a tree-like topology without providing a statistical measure for how reliable the resulting fit is.

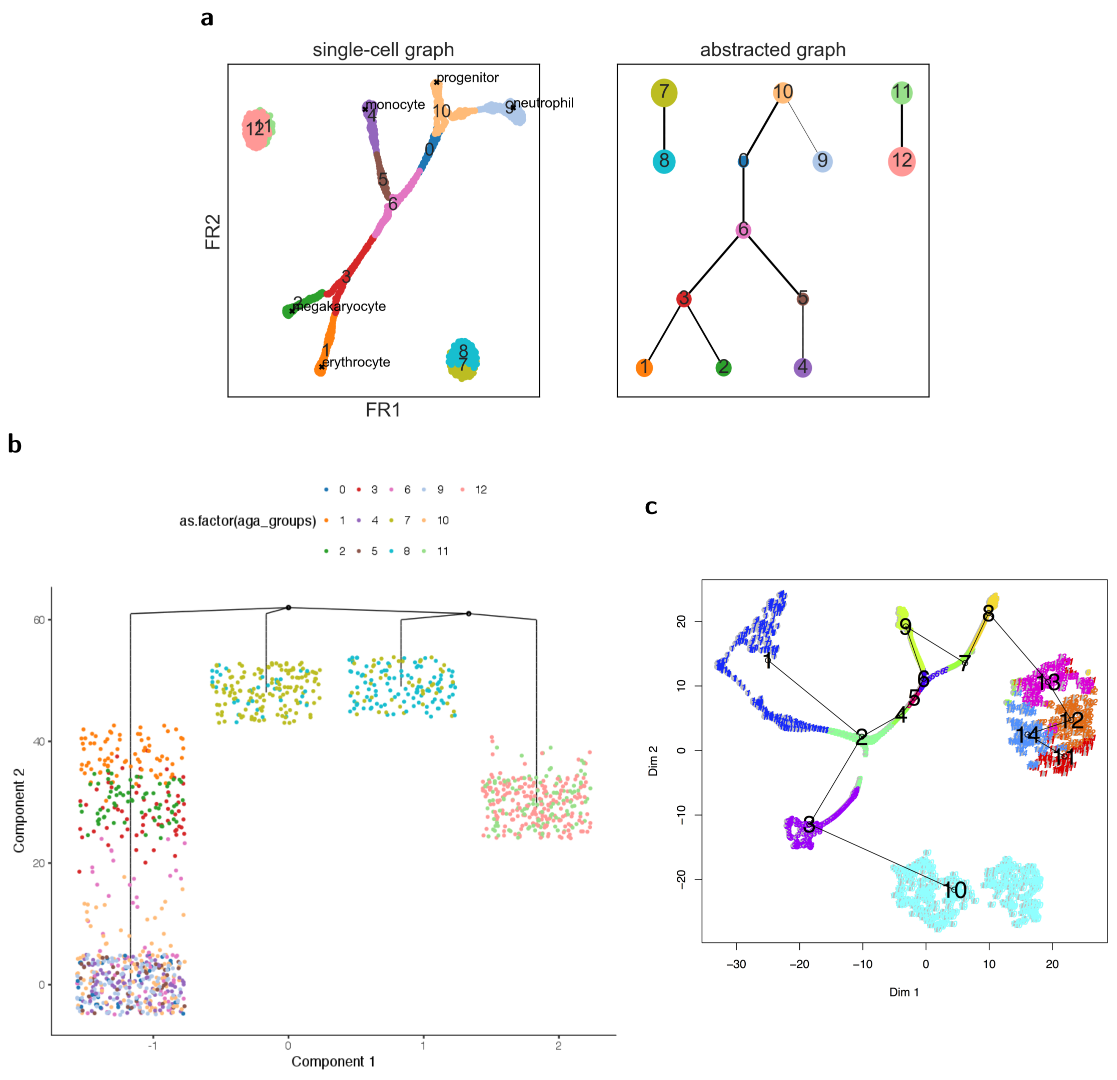
Spade [11], StemID 2 [13], Eclair [12], TSCAN [32] and Mpath [33] use different clustering algorithms such as k-means, k-medoids, hierarchical clustering or DBSCAN in a dimensionality-reduced space. In a second step, they fit a minimum spanning tree to either the centroid or medoid distances or to projections of cells on linear connections between centroids or medoids. In this, distances are computed using simple, fixed distance measures such as the euclidean or the correlation-based distance. Neither do these distances between clusters measure how well and if clusters are connected with each other, nor do these methods try to invoke a statistical model to address this question. The computationally expensive sampling procedures in StemID 2 and Eclair only partially alleviate the principle problem of high non-robustness that is caused by these deficiencies. Projections on linear connections between clusters assume a linear geometry of differentiation trajectories, which is certainly violated in practice. Hence, Mpath, for example, has only been shown to reconstruct processes with a single branching [33]. Moreover, it is important to note that none of the used clustering algorithms in these methods guarantees a topology preserving coarse-graining of the data: disconnected regions of data might cluster together and connected regions might be torn apart.

DPT [8] computes a random-walk based pseudotime for all cells. It cannot handle data with disconnected structure and is only able to detect single branchings which, in addition, is prone to violating the topological structure in the data (see, e.g., Figure 2c of Reference [10]). This problem becomes particularly pronounced in the extension of DPT to multiple branchings [22].

Rizvi *et al.* [10] use topological data analysis, in particular, the MAPPER algorithm [30], to construct a pseudotemporal ordering of cells. MAPPER constructs a partial coordinatization of the data that in the form of a simplicial complex, which has some similarity with the abstracted graph introduced in the present work. Both MAPPER's simplicial complex and the abstracted graph are graphs whose nodes correspond to clusters in the high-dimensional dataset and whose edges indicate connectivity of the clusters. However, the construction of MAPPER's simplicial complex differs fundamentally from the abstracted graph. In particular, the clusters do not correspond to regions of simple topology with high intra-connectivity, which can often be meaningfully interpreted as cell types. Hence, in contrast to graph abstraction, MAPPER does not generate an easily interpretable partitioning of the data into connected and disconnected regions, but a highly fine-grained, overlapping clustering, where clusters merely serve a technical purpose. Thereby, the algorithm can only be meaningfully applied to connected data. Moreoever, MAPPER does not provide the robust, random-walk based distance measure, which we use as a continuous coordinate for locating single cells within the abstracted graph. Even though topological data analysis allows the definition of continuous coordinates on the simplicial complex, their robustness and interpretability has not been shown. Additionally, MAPPER is *not* based on simplifying a k-nearest-neighbor graph of data points. We interpret graph abstraction as a pragmatic, easily-interpretable, scalable and robust way of performing topological data analysis for potentially disconnected data.

We do not compare our method to Wishbone [7], which can only detect a single branching, nor to the fundamentally different, fully supervised approach STEMNET [26].

bioRxiv preprint first posted online Oct. 25, 2017; doi: http://dx.doi.org/10.1101/208819. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.



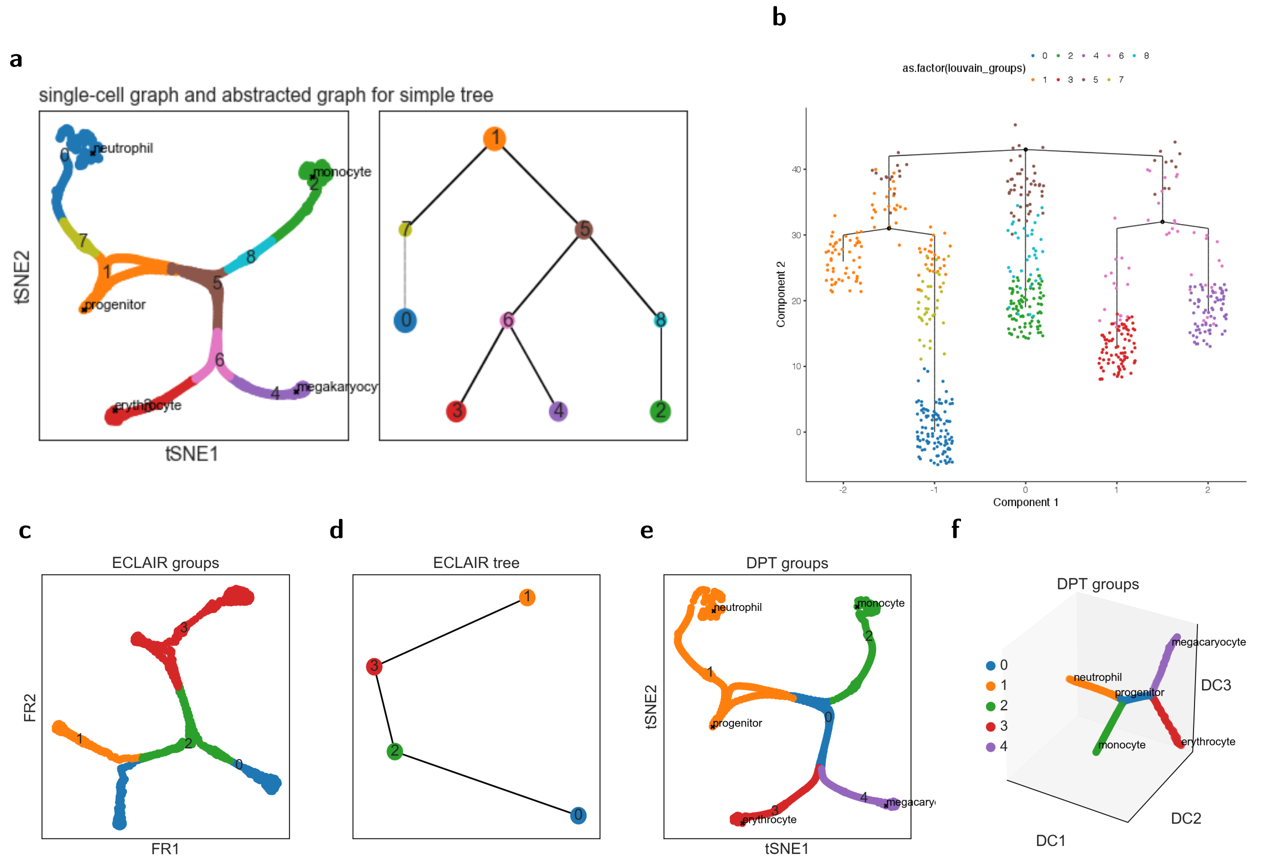
Supplemental Figure 1 | Comparison with Monocle 2 and stemID 2 for minimal example of simulated myeloid differentiation and clusters. a, Prediction of graph abstraction, reproduced from Supplemental Figure 6d. **b,** Prediction of Monocle 2 [6], the best result after testing several parameters for the latent-space dimension. The clusters (groups 7, 8 and 11, 12 in panel a) dictate the shape of the inferred tree, being responsible for three of the four observed branches. The continuous manifold is not resolved at all. The same coloring as in panel a is used. **c,** Prediction of the lineage tree of stemID 2, the successor of stemID [13]. The author of stemID, D. Grün, kindly provided parameters for the simulation. The inferred lineage tree shows a single branching at group 2 into groups 1 and 10, instead of the four branchings seen in Figure 2a. The coloring and numbering of groups is chosen internally by stemID 2.

Supplemental Note 1.2: Comparisons for minimal examples with known ground truth

We consider a minimal example with known ground truth to show that graph abstraction overcomes qualitative conceptual problems in the design of algorithms for the inference of lineage trees. The dataset consists in a connected tree-like manifold and two disconnected clusters and has a clearly defined ground truth — a computational model for hematopoiesis — and very little noise (Supplemental Note 6.1, Supplemental Figure 6.1). Nonetheless, none of Monocle 2, StemID 2, Eclair and DPT produce sensible results. Only when we removed the clusters from the data, these algorithms made sensible predictions. To reproduce the following comparisons and to get more information follow this link.

Graph abstraction recovers the ground truth (Supplemental Figure 1a). Monocle 2 [6] — even after testing several values for the latent-space dimension in Monocle 2 [6] — fits a tree to the

bioRxiv preprint first posted online Oct. 25, 2017; doi: http://dx.doi.org/10.1101/208819. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.



Supplemental Figure 2 | Comparisons for simulated myeloid differentiation giving rise to a simple tree-like manifold. Results using, **a,** graph abstraction, **b,** Monocle 2 [6], **c, d,** ECLAIR [12] and **e, f,** DPT [8] in its hierarchical implementation [22].

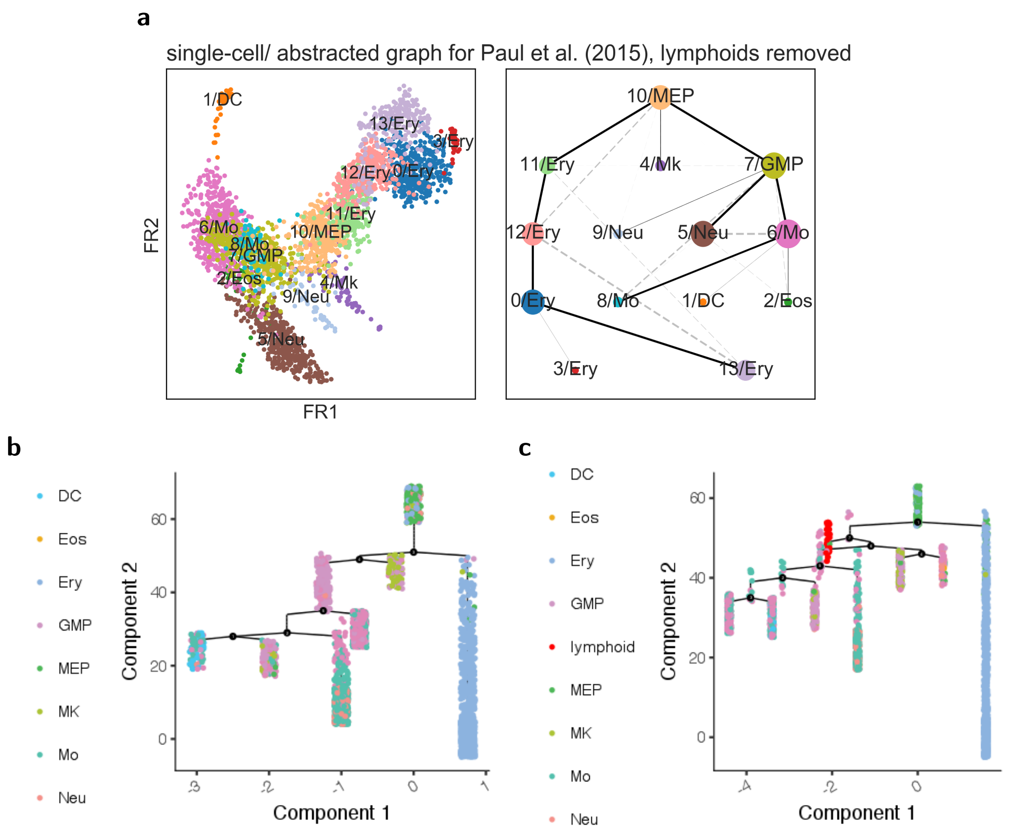
clusters and misses to recognize the continuous manifold in the data (Supplemental Figure 1b). D. Grün kindly provided parameters for running stemID 2 — the successor of stemID [13] — on the minimal example. The produced lineage tree though displays only a single of the three branchings (Supplemental Figure 1a) and erroneously connects the clusters and the manifold. For the minimal example, we could not produce any sensible result neither with Eclair [12] — even after optimizing parameters in correspondence with the author G. Giecold — nor DPT [8].

As a control, we aimed to obtain sensible results with the competing algorithms and considered a simpler dataset that only contains the continuous tree-like manifold of the previous example. Graph abstraction recovers the ground truth (Supplemental Figure 2a). Monocle 2 can be tuned — by adjusting the latent space dimension — to yield the correct result (Supplemental Figure 2b). Eclair [12] obtains a wrong result even for this simple tree (Supplemental Figure 2c, d). DPT [8] does, by construction, not infer a lineage tree but merely detects two branching subgroups; similar to a clustering algorithm. In a hierarchical implementation [22], it detects an arbitrary number of groups. Using the latter to detect four branchings we can detect two branchings (Supplemental Figure 2e) but fail to detect a third. Note that only when using diffusion maps for visualization, the clustering of groups appears natural (Supplemental Figure 2f).

Supplemental Note 1.3: Comparison for data of Paul *et al.* [18]

In the recent Monocle 2 paper of Qiu *et al.* [6] the data of Paul *et al.* [18] served as an example for the reconstruction of a complicated differentiation tree in Supplemental Figure 16. In the preprocessing step for the analysis of this data, Qiu *et al.* removed a cluster of lymphoid cells. In many situations, clusters of cells might not be annotated or not be clearly disconnected and it might not be clear

bioRxiv preprint first posted online Oct. 25, 2017; doi: http://dx.doi.org/10.1101/208819. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.



Supplemental Figure 3 | Comparison with Monocle 2 for data of Paul *et al.* [18]. a, Running the exact same parameters as in Figure 2a, but after removing the lymphoid cells, as done by Qiu *et al.* [6]. The resulting abstracted graph is consistent with Figure 2a. **b,** Monocle 2's multiple branching example of Supplemental Figure 16 of Reference [6] using the same color coding as in the original publication. **c,** Rerunning Monocle 2 with the exact same parameters as for panel b, but keeping the lymphoids as in Figure 2a. The resulting tree changed dramatically and is no longer biologically meaningful. For example, the lymphoid cells are placed in the myeloid differentiation and myeloid progenitors (GMP) and monocytes (Mo) are distributed over all terminal states. As Monocle 2 does not provide confidence measures, the user erroneously expects all results to predicted with high confidence.

whether one should remove them from the data. We therefore wondered what would happen when rerunning Monocle 2 with the exact same settings on the same data but keeping the cluster of lymphoids. While graph abstraction produces consistent results irrespective of the presence of this cluster (Supplemental Figure 3a, b), Monocle 2's inferred tree changed dramatically and displays qualitatively wrong biology, for instance, by placing the lymphoid cluster in the center of the myeloid differentiation. To reproduce the comparison and to get more information follow this link.

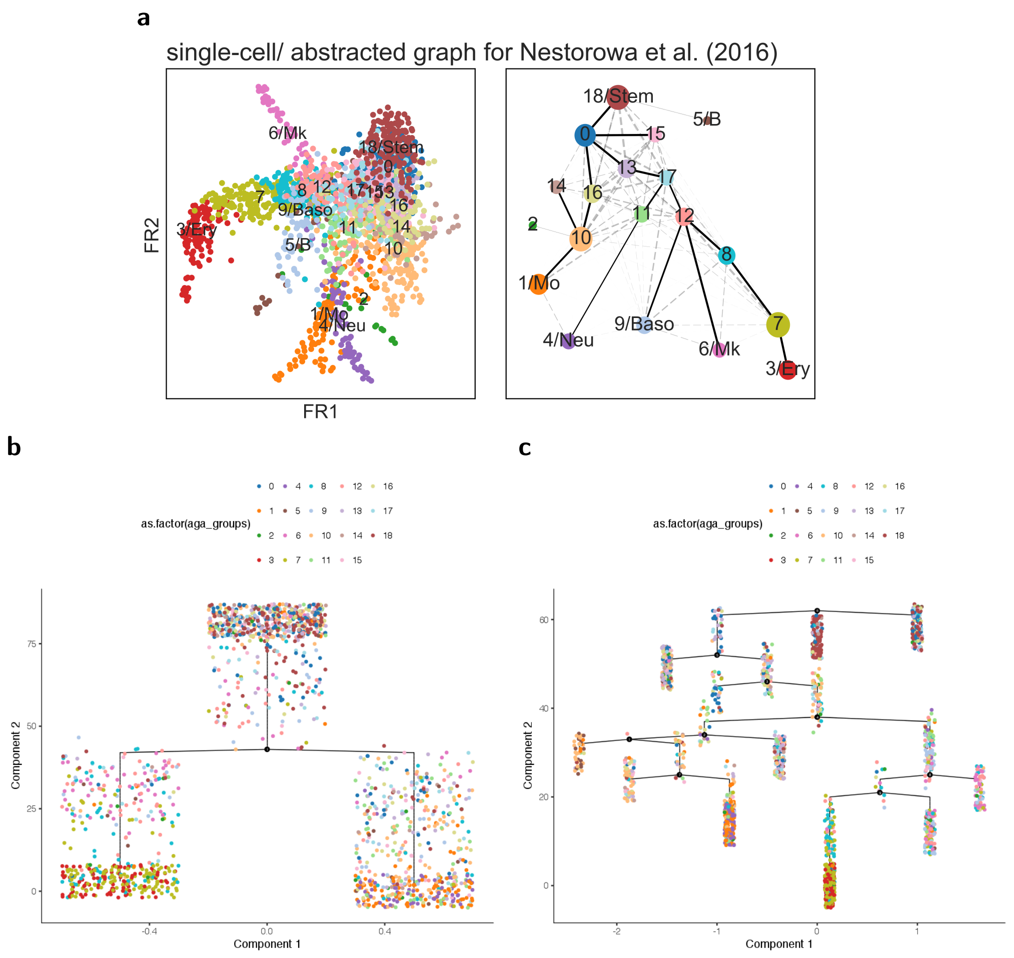
Supplemental Note 1.4: Comparison for data of Nestorowa *et al.* [19]

Supplemental Figure 4 shows a comparison for data of Reference [19]. To reproduce the comparison and to get more information follow this link.

Supplemental Note 1.5: Comparison of runtimes

Computing the result of Figure 1a using graph abstraction took 0.5 s. The stemID computation, including tSNE, ran for 17 min. Comparing this with 0.5 s for graph abstraction and 2.8 s for tSNE, both in Scanpy [22], graph abstraction is 309 times faster. The Monocle 2 computation took 13.8 s in the fastest case. Comparing this with 0.5 s for graph abstraction, graph abstraction is 28 times faster. The authors of Monocle 2 report a runtime of 9 min for 8 000 cells [34] and a linear

bioRxiv preprint first posted online Oct. 25, 2017; doi: http://dx.doi.org/10.1101/208819. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.



Supplemental Figure 4 | Comparison with Monocle 2 for data of Nestorowa *et al.* [19]. **a,** Result of graph abstraction as in Figure 2c, but colored by cell type annotation. **b,** Running Monocle 2 for a latent space dimension of 4 underestimates the complexity of the differentiation manifold. **c,** Running Monocle 2 for a latent space dimension of 10 recovers the expected biology that late erythrocytes (3/Ery) and megakaryoctes (6/Mk) appear in the same region of the tree. Nonetheless, there are qualitative inconsistencies: neutrophils (4/Neu) and monocytes (1/Mo) appear in the same terminal branch. Megakaryocytes (6/Mk) appear in two branches. Basophils (9/Baso) appear as progenitors of erythrocytes and megakaryoctes and the disconnected B cells appear within the tree. Monocle 2's tree-like representation of the data blurs the fact that the geometry of the data allows in fact many different paths through the data.

scaling. Extrapolation yields 76.5 min for 68 000 cells for which graph abstraction takes 2 min, which corresponds to a speedup of 38.

Supplemental Note 2: Random walks on graphs

On the single-cell level, the continuity of connections are believed to be well parametrized by a "pseudotime" [2, 3] that measures the distance covered in a continuous progression along a manifold. A robust kernel-based measure that can be easily extended to a graph, diffusion pseudotime, has recently been proposed by Haghverdi *et al.* [8]. This measure and similar scale-free random-walk based measures though do not account for clustering structure in the data; they are undefined for disconnected graphs. Below, we show how to overcome this limitation by extending these measures.

bioRxiv preprint first posted online Oct. 25, 2017; doi: http://dx.doi.org/10.1101/208819. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.

Supplemental Note 2.1: Interpreting random walks and their path distributions

It is important to note that in the whole paper, when we say "random walk on a graph", we mean a discrete-space Markov process on the state space given by the nodes of the graph and non-zero transition probabilities between any two connected nodes.

Such random walks can be used to probe the global topology of the single-cell graph $\mathcal{G}$ but do not provide a good model for the biological processes that one might hypothesize to have generated the data in the first place. The primary deficiencies of the Markov random walk when seen as a model for a biological process are the following.

- Undirectedness. When progressing along a differentiation trajectory, at some point, one expects commitment of a cell to a specific fate and a directed motion to that fate with some fluctutations. By contrast, the diffusive motion induced by the Markov random walk is highly non-directed, which leads to unrealistic paths that go back and forth and pass through remote regions of the graph.
- Independence of the expression of specific genes. The random walk is independent of the expression of specific genes; it only depends on global differences in the transcriptome. But, specific genes are known to play a role during commitment.

These deficiencies of the random walk become apparent already when modeling a biological process using the simple stochastic differential equation based model discussed in Supplemental Note 6.1.

The distribution of single-cell paths that correspond to a path through the abstracted graph, by contrast, resolves the problem of undirectedness by bounding the distribution to the ribbon of the connected sequence of groups.

Supplemental Note 2.2: Existing random-walk based distance measures

For a single-cell graph $\mathcal{G}$ with n_{nodes} nodes and n_{edges} edges, consider the normalized graph laplacian [35, 36]

$$L = I - T, \quad T = D^{-1}A, \tag{1}$$

where I is the $n_{\text{nodes}} \times n_{\text{nodes}}$ identity matrix and T is the transition matrix of the same shape. T is obtained from the weighted adjacency matrix A of $\mathcal{G}$ by normalizing with row sums of A, that is, D is the diagonal matrix that stores the degree of each node in $\mathcal{G}$. In practice, we compute the weights of the adjacency using a Gaussian decay with euclidian distance between two data points in gene expression space, see e.g. Reference [8]; after that, we density-normalize obtained weights [37, 38] as in Reference [8].

For a study of random walks generated by T, a spectral analysis of L and T is convenient and one hence considers the matrices $\widetilde{L}$ and $\widetilde{T}$, which are obtained by multiplying (1) with $D^{-\frac{1}{2}}$ from the left and with $D^{\frac{1}{2}}$ from the right

$$\widetilde{L} = I - \widetilde{T}, \quad \widetilde{T} = D^{\frac{1}{2}}TD^{-\frac{1}{2}}. \tag{2}$$

Hence, L and $\widetilde{L}$ have the same spectrum $\{1-\lambda_1, 1-\lambda_2, \dots\}$ and the spectrum of T and $\widetilde{T}$ is given as $\{\lambda_1, \lambda_2, \dots\}$ with $\lambda_1 = 1, \lambda_2 < \lambda_1, \dots$ for a connected graph $\mathcal{G}$ [35, 36]. For a disconnected graph with n_{comps} disconnected components, the adjacency matrix A has block-diagonal form with n_{comps} blocks and there are n_{comps} eigenvalues λ_i with value 1 and corresponding eigenvectors that are the indicator vectors of the connected components. The eigenvectors $\widetilde{v}$ of $\widetilde{T}$ are related to the right eigenvectors v of T as [35–37]

$$v_{r\iota} = \frac{\widetilde{v}_{r\iota}}{\sqrt{D_{\iota\iota}}} \qquad \forall r, \iota. \tag{3}$$

bioRxiv preprint first posted online Oct. 25, 2017; doi: http://dx.doi.org/10.1101/208819. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.

The right eigenvectors v of T are known as "diffusion map" coordinates [37], whereas the left eigenvectors span the space of probability distributions of configurations of the Markov process. The first right eigenvector, corresponding to $\lambda = 1$, is the all-one vector — with only 1 as entry — and the first left eigenvector is the stationary state of the Markov process.

Using this notation, one obtains the mean commute time — the average number of steps one needs to arrive from node ι_1 to another node ι_2 — in equation (4a) [35]. One obtains "diffusion distance" [37, 39] in equation (4b) and "diffusion pseudotime" [8] in equation (4c).

$$\text{mean commute time}(\iota_1, \iota_2) = 2n_{\text{edges}} \sum_{r=2}^{n_{\text{nodes}}} \left(\frac{1}{1-\lambda_i}\right)^2 (v_{r\iota_1} - v_{r\iota_2})^2, \tag{4a}$$

$$\text{diffusion distance}^{(n_{\text{steps}})}(\iota_1, \iota_2) = \sum_{r=2}^{n_{\text{nodes}}} \lambda_i^{2n_{\text{steps}}} (v_{r\iota_1} - v_{r\iota_2})^2, \tag{4b}$$

$$\widetilde{\text{dpt}}(\iota_1, \iota_2) = \sum_{r=2}^{n_{\text{nodes}}} \left(\frac{\lambda_i}{1-\lambda_i}\right)^2 (\widetilde{v}_{r\iota_1} - \widetilde{v}_{r\iota_2})^2 + (\widetilde{v}_{1\iota_1} - \widetilde{v}_{1\iota_2})^2, \tag{4c}$$

$$\text{dpt}(\iota_1, \iota_2) = \sum_{r=2}^{n_{\text{nodes}}} \left(\frac{\lambda_i}{1-\lambda_i}\right)^2 (v_{r\iota_1} - v_{r\iota_2})^2, \tag{4d}$$

$$\text{algebraic distance}^{(k)}(\iota_1, \iota_2) = \sum_{r=1}^{r_{\max}} (\chi_{r\iota_1}^{(k)} - \chi_{r\iota_2}^{(k)})^2, \quad \chi_r^{(k)} = L^k \chi_r^{(0)}. \tag{4e}$$

With equation (4d), we give a slightly altered definition of diffusion pseudotime, which is consistent with the other measures and was found to be equally-well performing for applications in single-cell biology by the authors of Reference [8] — note that using the v_r basis instead of $\widetilde{v}_r$, the last term in equation (4c) becomes zero. Highly related is algebraic distance on the graph as given in (4e) [see e.g. 38].

Supplemental Note 2.3: Interpretation of random-walk based distance measures

Random-walk based distances on graphs have first been used to cluster graphs in Reference [39] (4b) and Reference [40] (4a), albeit without considering neighborhood graphs of data points. Reference [37] proposed "diffusion distance" for measuring the similarity between data points, albeit not on a graph, but for a Gaussian kernel matrix. Then, a random-walk based distance measure for single-cell data has first been proposed to measure the similarity between cells by Reference [8]; again not formulated for graphs. These authors introduced the measure of equation (4c), which integrates out the number of steps n_{steps} in (4b) to arrive at a scale-free measure.

The dpt measure is highly similar to (4a), which is easier to interpret and scale-free, too: it measures the average number of steps it takes to walk from ι_1 to ι_2. While equation (4b) arises as the summed difference of transition probabilities to all other nodes for two random-walks of length n_{steps} that start at nodes ι_1 and ι_2, respectively [37, 39], (4d) considers the sum over all numbers of n_{steps}, hence a difference of "accumulated transition probabilities", which are difficult to interpret; the interpretation of equation (4c) is not easier.

Algebraic distance, which has been used for graph partitioning in recent years [38], approximates (4a) and diffusion pseudotime and provides the computationally most efficient way of computing a random-walk based distance measure.

Supplemental Note 2.4: Random-walk based distance measures for disconnected graphs

Evidently, both scale-free distance measures, mean commute time (4a) and diffusion pseudotime (4c), are not defined for a disconnected graph $\mathcal{G}$ for which $n_{\text{comps}} > 1$ eigenvalues are 1: they yield

bioRxiv preprint first posted online Oct. 25, 2017; doi: http://dx.doi.org/10.1101/208819. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.

an infinite distance even for two nodes ι_1 and ι_2 that are in the same connected component of $\mathcal{G}$. It is important to realize that each connected component of $\mathcal{G}$ automatically leads to a block T_b in the transition matrix T that is itself a valid transition matrix and the spectrum of T is the union of the spectra of the blocks T_b. The eigenvectors of T are the eigenvectors of the blocks T_b filled with zeros at the positions of the other blocks [see e.g. 36]. Hence, we propose to extend mean commute time and diffusion pseudotime for disconnected graphs as

$$\text{mean commute time}(\iota_1,\iota_2) = 2n_{\text{edges}} \sum_{r=n_{\text{comps}}+1}^{n_{\text{nodes}}} \left(\frac{1}{1-\lambda_i}\right)^2 (v_{r\iota_1} - v_{r\iota_2})^2, \tag{5a}$$

$$\widetilde{\text{dpt}}(\iota_1,\iota_2) = \sum_{r=n_{\text{comps}}+1}^{n_{\text{nodes}}} \left(\frac{\lambda_i}{1-\lambda_i}\right)^2 (\tilde{v}_{r\iota_1} - \tilde{v}_{r\iota_2})^2 + \sum_{r=1}^{n_{\text{comps}}} (\tilde{v}_{r\iota_1} - \tilde{v}_{r\iota_2})^2. \tag{5b}$$

$$\text{dpt}(\iota_1,\iota_2) = \sum_{r=n_{\text{comps}}+1}^{n_{\text{nodes}}} \left(\frac{\lambda_i}{1-\lambda_i}\right)^2 (v_{r\iota_1} - v_{r\iota_2})^2, \tag{5c}$$

The distribution of zeros in the eigenvectors v_r and $\tilde{v}_r$ guarantees that for two nodes ι_1 and ι_2 in the same connected component b, only the spectrum of the block transition matrix T_b contributes. For two nodes ι_1 and ι_2 in two disconnected components, the measures take the sum of their maximum values in both components, which should be interpreted as infinite. Without problem, one can make this explicit in the equations by distinguishing cases in which ι_1 and ι_2 belong to the same component from cases in which they belong to different components.

We note that, in practice, instead of summing over all eigenvectors n_{nodes}, we sum over a low number of eigenvectors — "diffusion components" in the language of Coifman *et al.* [37] — as others [8, 40].

While in the present publication, we use equation (5b) throughout, we expect that equation (5a) could be useful in the future due to its easier interpretation.

Supplemental Note 3: Measuring connectivity of partitions of a graph

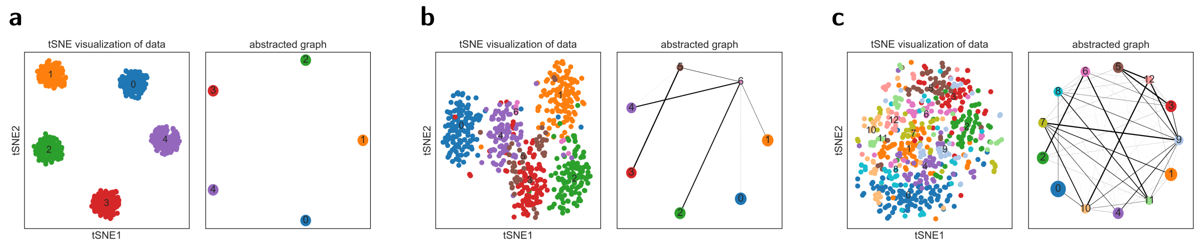
The simplest measure for connectivity of two partitions of $\mathcal{G}$ is the number of connecting edges between two partitions. By computing a statistic of this number, one can measure confidence in an actual connection between two partitions of $\mathcal{G}$, as opposed to a connection that is based on spurious edges (Supplemental Note 3.1). In the visualization of abstracted graphs, edge width is proportional to the confidence in the presence of an actual connection (see, e.g., Figure 1). This allows, for example, to assess how strongly data clusters into disconnected regions (Supplemental Figure 5). Another natural definition of connectivity, similar to ideas in hierarchical clustering on graphs [39], can be based on a random-walk based distance measure (Supplemental Note 3.2).

Note that topological data analysis uses clustering algorithms that lead to overlapping clusters and by that circumvents the definition of elaborate measures of connectivity: two clusters are connected if they have finite overlap. We do not adopt this approach as overlapping graph partitioning algorithms do not scale well to large datasets and their results are difficult to interpret when compared to the well-established Louvain algorithm for modularity optimization [41] that has been established for single-cell analysis [15].

Supplemental Note 3.1: Edge-statistics based measure for connectivity

Consider $n_{\text{partitions}}$ partitions $\{i\}$ of $\mathcal{G}$ and let the matrix $K = (k_{ij})_{i,j=1}^{n_{\text{partitions}}}$ count the number of inter- and intra-partition edges on $\mathcal{G}$. Hence, the nodes of partition i are involved in $k_i = \sum_j k_{ij}$ edges. The frequency of edges that connect to partition j is $\theta_j = \frac{k_j}{n_{\text{edges}}}$. For the null model of randomly connecting partitions, the expected frequency of connections between i and j is $\theta_i \theta_j$ [42].

bioRxiv preprint first posted online Oct. 25, 2017; doi: http://dx.doi.org/10.1101/208819. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.



Supplemental Figure 5 | Assessing the degree to which data clusters by measuring connectivity of partitions. Here, we show samples from three Gaussian mixture models **(a, b, c)**, which display different degrees of clustering structure: the number of centers is fixed to 5 but the standard deviation is increased from 1 (a) over 6 (b) to 10 (c). Graph abstraction measures how well Louvain-clustering based partitions [41] separate from each other. For standard deviation 10 (c), the clustering result predicts fully connected partitions, hence a dense abstracted graph $\mathcal{G}^*$. The geometry of data forms a fully continuous structure.

As in the definition of modularity [42], we compare the expected frequency $\theta_i\theta_j$ to the observed frequency k_{ij}/n_{edges} of the number of edges.

$$M_{ij} = K_{ij}/n_{\text{edges}} - \theta_i\theta_j \tag{6}$$

Other than in modularity optimization, here, we consider the distribution of M_{ij} to obtain a p-value for rejecting the null-hypothesis of that i and j are connected by spurious edges. The number of connections is the sum of n_{edges} Bernoulli variables with mean $\theta_i\theta_j$. If edge numbers are high enough, one can use the central limit theorem and assume a Gaussian with variance $\theta_i\theta_j(1-\theta_i\theta_j)/n_{\text{edges}}$ leading directly to a p-value without costly sampling.

$$\begin{aligned} \mathrm{E}\left[M_{ij}\right] &= 0 \\ \mathrm{var}\left[M_{ij}\right] &= \theta_i\theta_j(1-\theta_i\theta_j)/n_{\text{edges}} \end{aligned} \tag{7}$$

More refined tests could be developed, but our tests confirmed that this computationally cheap setup works well in many settings.

Supplemental Figure 6 illustrates the confidence in connections of clustering data, where clusters have different degree of connectivities.

Supplemental Note 3.2: Random-walk based measure for connectivity

Besides measuring connectivity between partitions using the number of connecting edges one can also use the random-walk based distance measure d, discussed in Supplemental Note 2.

The distance d strongly correlates with the number of paths between two points [see, e.g., 37] and hence allows to robustly define connectivity between partitions as measured by the number of the connecting paths between their nodes. Measures for the connectivy of two partitions i_1 and i_2 can then be obtained by invoking simple summary functions for the nodes $\{\iota_1\}$ and $\{\iota_2\}$ in these partitions

$$c_1(i_1,i_2) = \min_{\iota_1\in P_1,\iota_2\in P_2} d(\iota_1,\iota_2), \tag{8a}$$
$$c_2(i_1,i_2) = \mathrm{avg}_{\iota_1\in P_1,\iota_2\in P_2} d(\iota_1,\iota_2), \tag{8b}$$
$$c_3(i_1,i_2) = \mathrm{median}_{\iota_1\in P_1,\iota_2\in P_2} d(\iota_1,\iota_2), \quad i_1,i_2\in\mathcal{G}^*,\ \iota_1,\iota_2\in\mathcal{G}. \tag{8c}$$

Similar summary functions appear in different flavors of hierarchical clustering. Each of these functions comes with different advantages and disadvantages. Taking the minimum is independent of the specific shape of a partition but is prone to outliers: it is only a viable option as the distance measure d itself is highly robust being computed as an average over all random walks on the graph. Taking the average or median is robust but can cause perfectly attached partitions of the graph

bioRxiv preprint first posted online Oct. 25, 2017; doi: http://dx.doi.org/10.1101/208819. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.

appear far way if they are long-stretched out, that is, if d increases strongly when walking through them.

As even in the long-established hierarchical clustering algorithms, no definitive answer for the best choice of summary function has been found, we also do not claim to give a definitive answer. In practice, we use $c^{\text{rw}} = c_1$. For the case in which, for a given partition i, we want to determine to which partition $j \neq i$ is most strongly connected, we adapt the following heuristic: if c_1 yields no significant answer but c_3 does, we use c_3 as a decision criterion, and otherwise c_1.

While the random-walk based measure for connectivity has the advantage of directly measuring the continuity of paths along the graph $\mathcal{G}$, it is much harder — and might be impossible — to establish a statistical model for measuring confidence. To nonetheless be able to compare the connectivity of partitions in a tree-like subgraph $\mathcal{T}^*$ of $\mathcal{G}^*$, which we assume to better capture the continuous structure of the data, we use the following heuristic. We compute the median connectivity $c^{\text{tree}} = \text{median}_{(i,j)\in\mathcal{T}^*} c(i,j)$ and assume an exponentially decaying confidence in the presence of an actual connection:

$$q(i,j) = \begin{cases} 1 & \text{if } \frac{c^{\text{tree}}}{c(i,j)} < 1, \\ \exp\left(-\left(\frac{c^{\text{tree}}}{c(i,j)} - 1\right)\right) & \text{else.} \end{cases} \tag{9}$$

We tested that a Gaussian decay yields qualitatively comparable results.

Supplemental Note 4: Partitioning the single-cell graph

At the heart of graph abstraction lies the assumption that the single-cell graph $\mathcal{G}$ — the k-nearest-neighbor graph of data points — provides a powerful representation of data. This assumption is on one hand based on the community's success with graph-based clustering [15, 17, 41], pseudotime inference [8], visualization [16, 23] and t-distributed stochastic neighborhood embedding [tSNE, 43, 44] — the latter can be seen as using a fully-connected graph with decaying weights. On the other hand, it is based on the observation that neighborhood graphs robustly generalize any local distance measure to a global scale. As any fixed distance measure can at best encode a very rough notion of biological similarity with an exploding error for large distances, it is more robust to only evaluate it locally, and construct the global disttances from the graph of neighborhood relations.

Here, we only consider established preprocessing recipes [16, 20, 21] for single-cell transcriptomic data, each of which give rise to a different fixed distance measure. For single-cell imaging data. we consider a learned distance measure as induced by the feature space of a deep learning model [29]. Any other distance measure, for example, the kernel-based measure of Reference [45], would also be a viable option.

A partitioning of $\mathcal{G}$ that maximizes the ratio of intra- to inter-partition edges is natural in the sense that it reveals regions of the graph with different connectivity and hence, different topology. Optimizing this ratio is known as optimizing modularity [42]. An efficient algorithm for this [41] has been suggested for single-cell biology by Reference [15]. Loosely speaking, one obtains the clearest coarse-grained picture of the data at a fixed resolution if choosing the partitioning that maximizes modularity. While the original implementation of the Louvain algorithm Reference [41] contained an error that might lead to disconnected partitions — hence violates the topological structure of the data — we use a highly efficient and correct implementation [46]. Note that the Louvain algorithm has also been adpoted by popular single-cell analysis toolkits such as Seurat [20] and Cell Ranger [21].

Many other possibilities for partitioning $\mathcal{G}$ — or clustering the data — exist. We highlight the graph-based hierarchical clustering of Reference [39], which is based on a random-walk based distance measure. Using points that maximize a similar distance measure, we extend this method to better capture topological properties of the graph that are encoded in the distribution of these maxima.

bioRxiv preprint first posted online Oct. 25, 2017; doi: http://dx.doi.org/10.1101/208819. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.

We refer to this method as "iterative matching of extremal points" (main text and Supplemental Note 8).

Supplemental Note 5: Reconciling clustering and pseudotime algorithms

Here, we provide a more formal explanation of the discussion of Figure 1 in the main text. The aim of any pseudotemporal ordering of given data is to reconstruct the continuous latent variable that associates with the variation in the data; presumably the process that generated the data. Furthermore, pseudotemporal ordering of cells enables the identification of the relative timing of different events during the process — it tries to represent the internal "clock" of cells as encoded in its molecular configuration. A clustering analysis, by contrast, relates neither cells nor clusters to each other. With the abstracted graph $\mathcal{G}^*$, which describes the connectivity — or "continuity" — relations $c(i, j)$ of clusters i and j of $\mathcal{G}$, and the pseudotime measure $d(\iota_1, \iota_2)$, which measures the continuous progression of a cell ι_1 to a cell ι_2, one reconciles the result of a clustering analysis with the aim of a pseudotime analysis: Each cluster is related to any other cluster as either being disconnected or connected with one or several paths of high confidence in $\mathcal{G}^*$. Moreover, within each cluster, each cell is ordered according to pseudotime. One can hence trace a continuous process from a root cell ι_{root} in a root cluster i_{root} to any terminal cell ι_{end} in its terminal cluster i_{end} by following a path of high confidence $(i_{\text{root}}, i_1, i_2, \ldots, i_{\text{end}})$ in $\mathcal{G}^*$. In each step of this path, the pseudotemporal ordering provides an ordering with single-cell resolution and, hence, one traces the progression of single cells along an ensemble of paths of high-confidence in $\mathcal{G}$. Thereyby, graph abstraction provides a topology preserving map of cells as $(\mathcal{G}^*, d)$. Without the abstracted graph $\mathcal{G}^*$, computing the ensemble of highly confident paths from from i_{root} to ι_{end} in $\mathcal{G}$ is a computationally much harder and an unsolved problem. The heuristics for its inference are less transparent and easy to control than the heuristics involved in partitioning a graph $\mathcal{G}$ and generating an abstracted graph $\mathcal{G}^*$.

Supplemental Note 6: Hematopoiesis

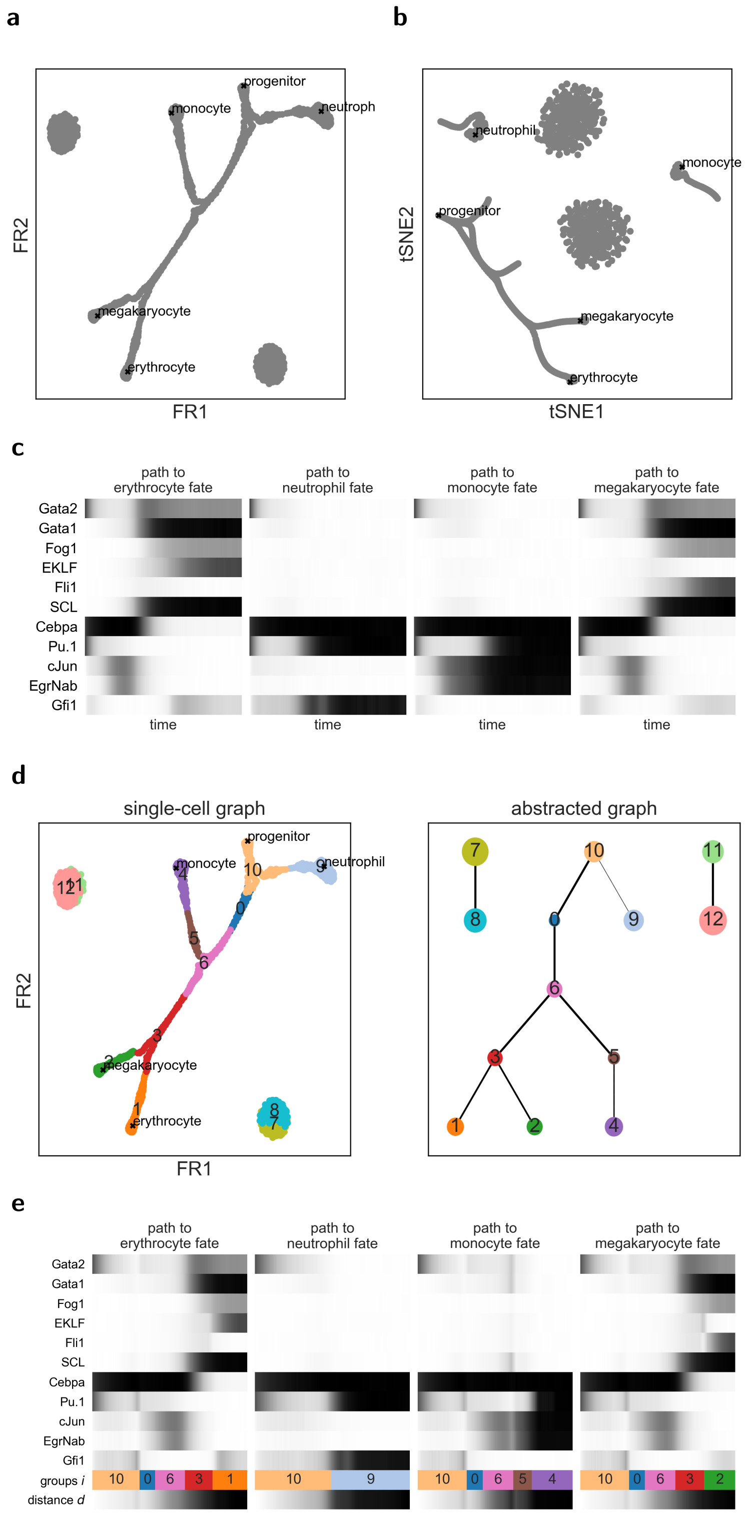
Supplemental Note 6.1: Simulations — a minimal dataset that contains a continuous manifold and disconnected clusters

We use a literature-curated qualitative – boolean – gene regulatory network of 11 genes that aims to describe myeloid differentiation [47] and has been used for benchmarking the reconstruction of gene regulatory network from a single-cell graph of state transitions in Reference [48]. The boolean network evolves according to

```
Gata2 = Gata2 and not (Gata1 and Fog1) and not Pu.1,
Gata1 = (Gata1 or Gata2 or Fli1) and not Pu.1,
Fog1 = Gata1,
EKLF = Gata1 and not Fli1,
Fli1 = Gata1 and not EKLF,
SCL = Gata1 and not Pu.1,
Cebpa = Cebpa and not (Gata1 and Fog1 and SCL),
Pu.1 = (Cebpa or Pu.1) and not (Gata1 or Gata2),
cJun = Pu.1 and not Gfi1,
EgrNab = (Pu.1 and cJun) and not Gfi1,
Gfi1 = Cebpa and not EgrNab.
```

These boolean equations are translated into ordinary differential equations following Reference [49]. Within Scanpy [22], they are simulated as stochastic differential equations by adding Gaussian noise.

bioRxiv preprint first posted online Oct. 25, 2017; doi: http://dx.doi.org/10.1101/208819. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.



Supplemental Figure 6 | Approximate graph abstraction for minimal example of simulated myeloid differentiation and clusters. a, Fruchterman-Reingold (FR) visualization. **b,** t-SNE visualization. **c,** Four representative realizations of time series for simulated myeloid differentiation. **d,** Single-cell graph colored by partitions and abstracted graph in which edge width is proportional to the confidence in the presence of the edge. **e,** Gene dynamics along the most probable paths to erythrocytes, neutrophils, monocytes and megakaryocytes in the abstracted graph. Within each group, cells are ordered according to pseudotime from the earliest progenitor cell.

Simulations result in four classes of realizations of gene expression time series, each of which corresponds to the convergence to an attractor that represents a certain cell fate of myeloid progenitors:

bioRxiv preprint first posted online Oct. 25, 2017; doi: http://dx.doi.org/10.1101/208819. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.

erythrocyte, neutrophil, monocyte and megakaryocyte (Supplemtal Figure 6). We concatenate four typical realizations (Supplemtal Figure 6c, d) with 160 time steps, which yields 640 data points in total.

To model clustering, we sample 640 data points from a Gaussian mixture model with two Gaussians and random centers in an 11-dimensional space. The minimal dataset of Figure 2 and Supplemental Figure 6 consists of the concatenated data matrices of the simulated myeloid progenitor development data and the Gaussian mixture model, corresponding to 1280 cells.

Supplemental Note 6.2: Data of Paul *et al.* [18] and Nestorowa *et al.* [19]

Supplemental Figure 7 provides additional visualizations for Figure 2.

Supplemental Note 7: Comparing graph topologies

The established algorithm for detecting the existence of an isomorphism between two graphs reveals an exact match of two graph topologies if nodes are unlabelled. But as the nodes of the abstracted graph $\mathcal{G}^*$ label partitions of $\mathcal{G}$, that is, sets of data points, we need another algorithm to compute the agreement of topologies: We are interested in whether the topology between two abstracted graphs $\mathcal{G}_1^*$ and $\mathcal{G}_2^*$ agree under the constraint that the node labels of $\mathcal{G}_1^*$ and $\mathcal{G}_2^*$ are consistent with each other. Moreover, instead of only detecting exact matches, we aim for a continuous measure of agreement.

Supplemental Note 7.1: Associating a partitioning with a reference partitioning

To establish such a measure, we first compute the overlaps of the partitions labelled by $\mathcal{G}_1^*$ and by $\mathcal{G}_2^*$ (Supplemental Figure 8a, b). By that, we generate non-unique associations between partitions, as visualized in an association matrix (Supplemental Figure 8c). The association matrix can ether be normalized with respect to the reference groups $\mathcal{N}_1^*$ (Supplemental Figure 8c) or with respect to the new groups $\mathcal{N}_2^*$, respectively (Supplemental Figure 8d). In order to obtain a symmetric score that measures how well two partitions mutually overlap — are mutually contained in each another — we consider the minimum of both normalizations — the "minimal overlap" — for each combination of groups $(i_1, i_2) \in (\mathcal{N}_1^*, \mathcal{N}_2^*)$. Supplemental Figure 8e colors each partition in $\mathcal{N}_1^*$ with the partition in $\mathcal{N}_2^*$ with which it has the largest minimal overlap.

Supplemental Note 7.2: Comparing paths in abstracted graphs

For each shortest path between two leaf nodes in $\mathcal{G}_2^*$, there is a shortest path between the associated nodes in $\mathcal{G}_1^*$. This enables to compare the two paths and to count the fraction of steps that are consistent among two paths. To measure the agreement of the topologies between two abstracted graphs, we compute the fraction of agreeing steps and the fraction of agreeing paths over all combinations of leaf nodes in two given abstracted graphs.

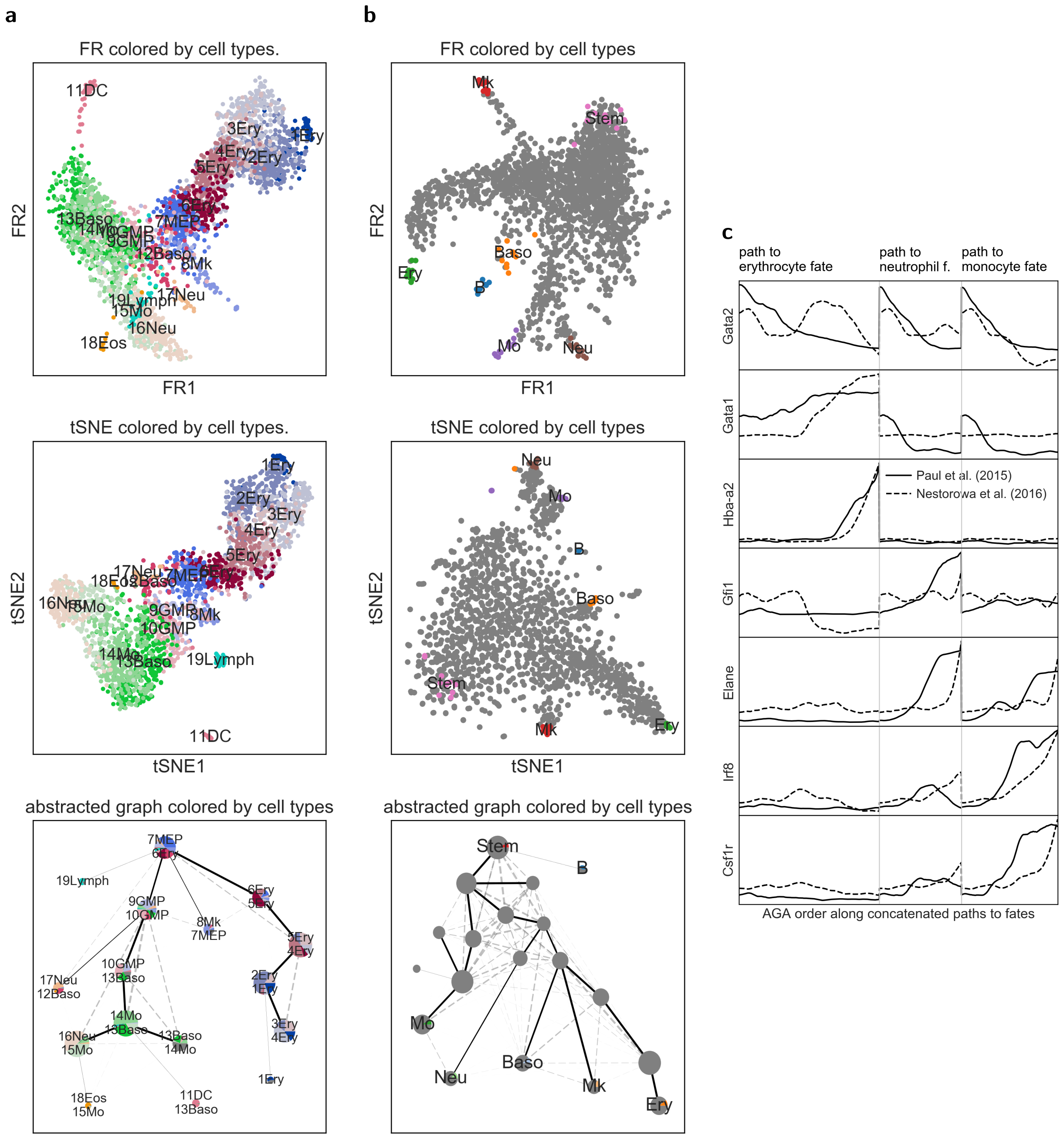
For instance, consider the shortest path between leafs (21, 2) in the reference graph $\mathcal{G}_1^*$ and the shortest path between leafs (7, 11) in the new graph $\mathcal{G}_2^*$ in Supplemental Figure 8a and b, respectively:

$$\begin{aligned} p_1 &= (21, 8, 18, 7, 9, 2), \quad p_1 \in \mathcal{G}_1^* \\ p_2 &= (7, 2, 9, 10, 11), \quad p_2 \in \mathcal{G}_2^*. \end{aligned} \tag{10}$$

By computing the overlap of reference partitions with new partitions, we can map p_1 to the label space of $\mathcal{G}_2^*$

$$p_1^{\text{mapped}} = ((7, 2), (6, 7, 2), (2, 7), (2, 9), (9, 10, 3), (11, 10)), \tag{11}$$

bioRxiv preprint first posted online Oct. 25, 2017; doi: http://dx.doi.org/10.1101/208819. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.

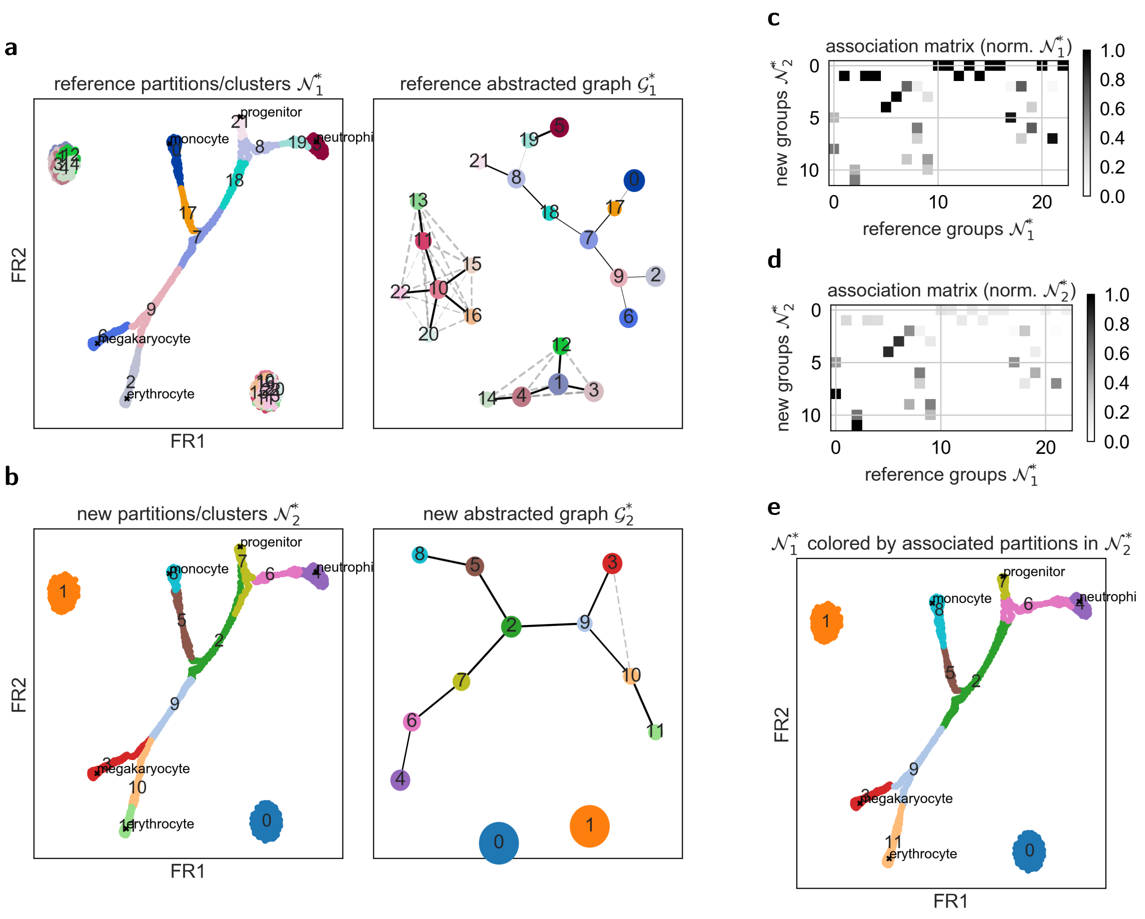


Supplemental Figure 7 | Additional visualizations for Figure 2. a, FR, tSNE and abstracted graph visualization of the data of Reference [18], colored by the cell type annotations of the original publication. The fractional overlaps of these cell type definitions and the partitions within graph abstraction are visualized as pie charts within the abstracted graph. **b,** The same for data of Reference [19], where we annotated differentiated cells as follows: Cells with RNAseq and index data were analyzed using SPRING [16], a tool for generating graph-drawing visualizations. Differentiated cells were annotated by selecting cells at the tip of each branch and identifying them using known marker genes. Hematopoietic stem cells that were gated as Lin- c-kit+ Sca-1+ CD34- Flk2- CD150+ CD48- EPCRhi cells. **c,** Comparative visualization of the gene dynamics along paths in Figure 2b and d. Data has been interpolated and normalized.

that is, partition 21 in $\mathcal{G}_1$ has finite minimal overlap with partitions 7 and 2 in $\mathcal{G}_2$, partition 8 in $\mathcal{G}_1$ has overlap with partitions 6, 7 and 2 in $\mathcal{G}_2$, and so on.

Transitioning through path p_2 and counting for each transition whether it's present or not in p_1^{mapped} allows to count the number of agreeing steps. If all steps agree with each other, the paths p_1 and p_2 agree with each other. In the example of equation (10), p_2 involves 4 steps, 4 of which agree with

bioRxiv preprint first posted online Oct. 25, 2017; doi: http://dx.doi.org/10.1101/208819. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.



Supplemental Figure 8 | Comparing topologies of abstracted graphs. a, b, Partitions obtained using Louvain clustering in two runs with different parameters, equivalent to those shown in Figure 2a: both abstracted graphs describe the same topology. Note that in Figure 2a, we use the Reingold-Tilford layout to draw the tree whereas here, we use the FR layout also for the abstracted graph. **c,** Reference partitions colored with the associated new partition that has the largest overlap.

p_1^{mapped}.

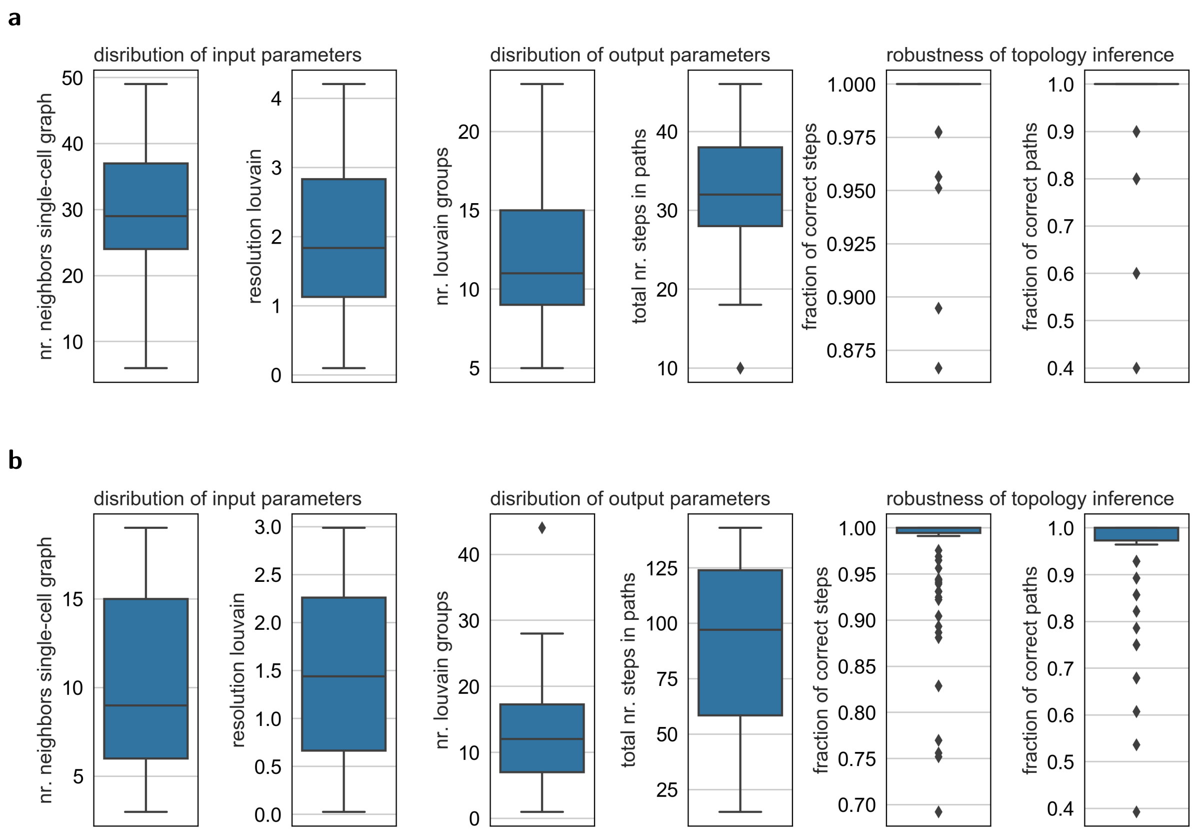
Supplemental Note 7.3: A related measure from the literature

Previously, it has been suggested to correlate the distribution of path lengths of all paths through trees as a measure for topological similarity of trees [12]. Specifically, for a tree whose nodes label sets of data points, the lengths of all paths between all pairs of data points are computed. The correlation of such path-length sets obtained for two trees is suggested as a measure for topological similarity of the two trees. Besides being highly redundant and costly to compute, the resulting measure is very rough as it does not map paths onto each other; that is, it does not account for inconsistencies of paths with the same length.

Supplemental Note 8: Iteratively constructing a topology-conserving tree

Consider the abstracted graph $\mathcal{G}^*$ in which edge weights measure how continuous partitions in the single-cell graph $\mathcal{G}$ are connected to each other. We aim to identify the tree-like subgraph $\mathcal{T}^*$ in the abstracted graph $\mathcal{G}^*$ that best reflects the global topology of $\mathcal{G}$. As $\mathcal{G}^*$ reflects the topology of $\mathcal{G}$ on a coarse resolution, it suffices to identify the tree $\mathcal{T}^*$ that best reflects the topology of $\mathcal{G}^*$. By this, we mean that walking along paths in $\mathcal{T}^*$ should best recover the paths in $\mathcal{G}^*$ and, by that, in $\mathcal{G}$.

bioRxiv preprint first posted online Oct. 25, 2017; doi: http://dx.doi.org/10.1101/208819. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.



Supplemental Figure 9 | Robustness of the inference of abstracted graphs in Figure 2. Sampling a wide variety of the two input parameters results in vastly varying numbers of partitions, hence vastly different clusterings of the data; note the large spread of the number of Louvain groups. Nonetheless, the topology is robustly inferred. We ran this robustness study for **a,** the minimal example and **b,** data of Reference [18] as in Figure 2a and b. Graph topologies are compared as explained in Supplemental Note 7.

In particular, walking along paths on $\mathcal{G}$ to points that stick out most, as measured by d, has to be reflected in the tree $\mathcal{T}^*$: its leafs need to contain extremal points. Note that if topologies are exactly reconstructed in $\mathcal{T}^*$, one can deform $\mathcal{T}^*$ by reconnecting the points in each node in $\mathcal{T}^*$ with points in neighboring nodes to recover the original graph $\mathcal{G}$.

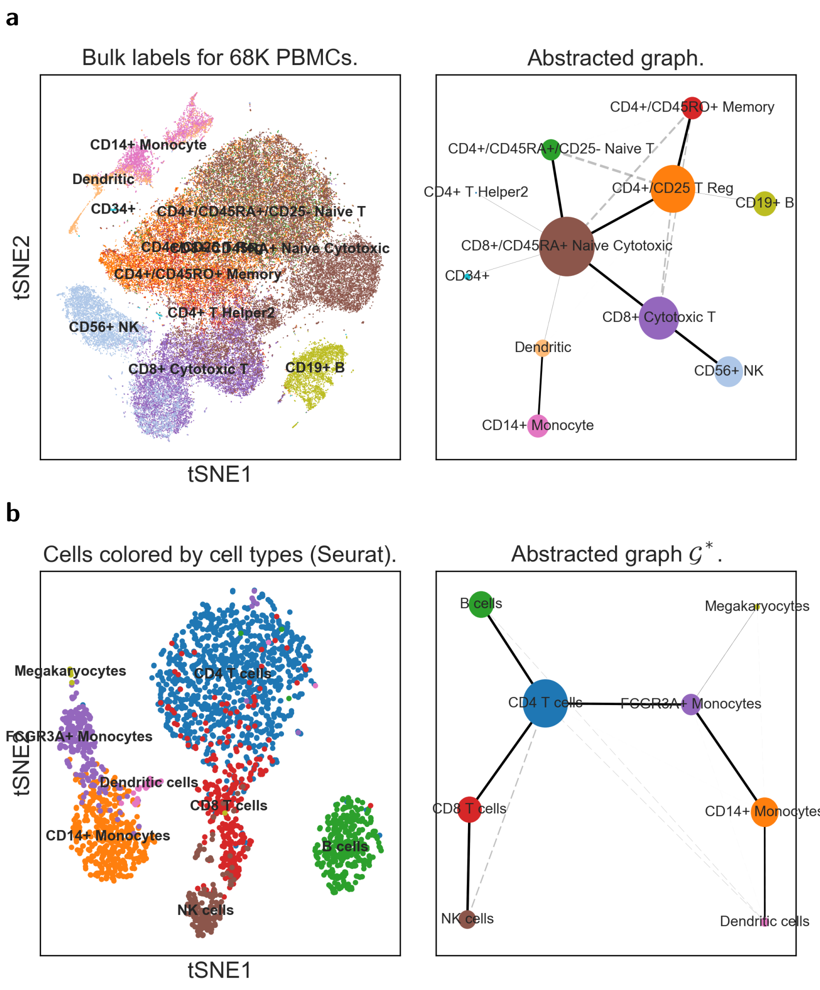
Supplemental Note 8.1: Minimum spanning tree

Finding the tree that maximizes connectivity and by that continuity between any two nodes is one way to address this problem without ever considering a set of extremal points. Finding the minimum spanning tree for inverse connectivity using an iterative, greedy algorithm like Prim's accomplishes this task. One grow's the tree by iteratively adding the node to a tree that maximizes connectivity.

Supplemental Note 8.2: Iterative matching of extremal points

A different approach to generating a topology-conserving tree is based on iteratively matching extremal points of $\mathcal{G}$ to leaf nodes of $\mathcal{T}$. We refer to this as "iterative matching". For simple cases, the results of MST and iterative matching agree. For more complicated cases, iterative matching tends to generate trees that rarely connect partitions that contain extremal points (Supplemental Figure 4). The interesting aspect of iterative matching is that, by construction, it constructs a hierarchical set of extremal points for all scales of the graph. While MST places "evident" extremal points on

bioRxiv preprint first posted online Oct. 25, 2017; doi: http://dx.doi.org/10.1101/208819. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.



Supplemental Figure 10 | Abstracted graphs for differentiated PBMCs. a, Reconstruction of motifs of the lineage tree of 68 000 PBMC cells [21]. This dataset serves as a negative control for the predictions of graph abstraction as it mostly lacks cells in the transitioning stages between different cell types. Hence, only motifs of the lineage tree of PBMCs are reconstructed in the abstracted graph. **c,** 3k PBMCs from 10X Genomics. Cells have been annotated using Seurat [20].

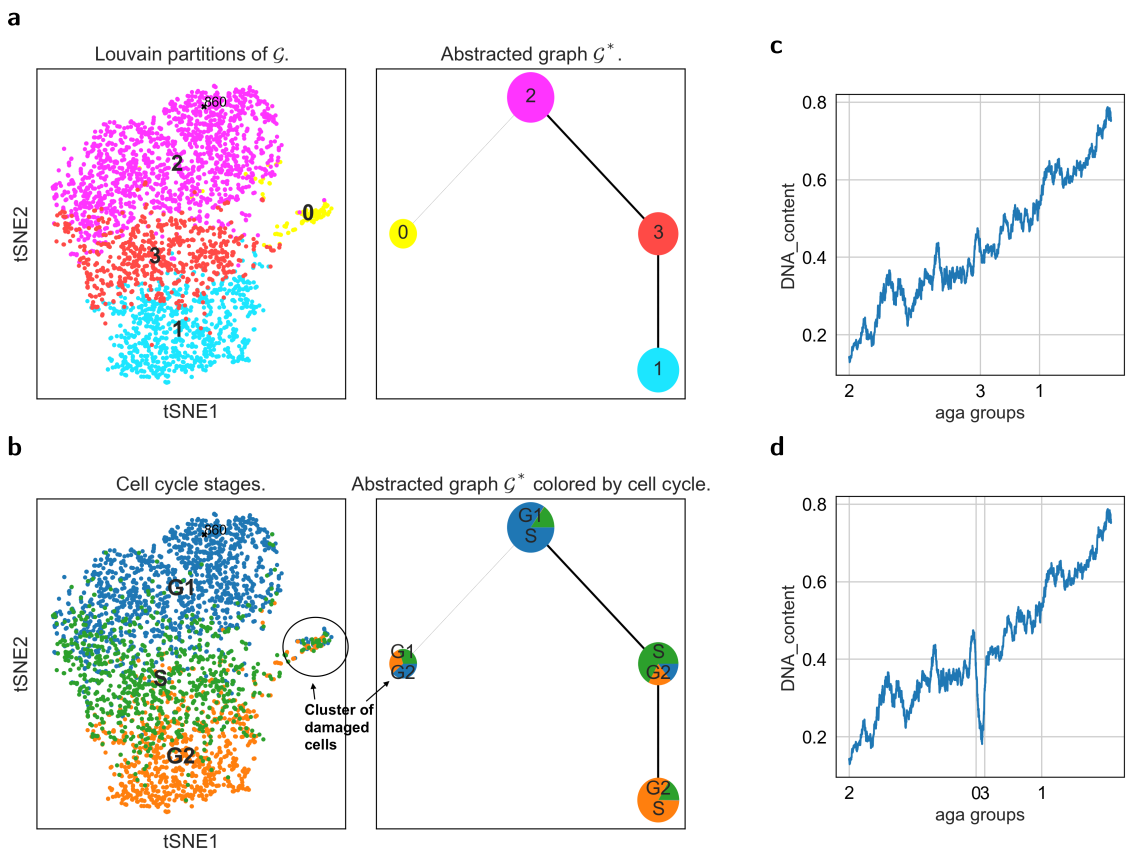
the largest scale in leafs of the tree, it might fail in doing this for smaller scales.[1] Finally, iterative matching can be used to generate partitions of $\mathcal{G}$ as in divisive hierarchical clustering.

Initializing the algorithm. Randomly select one of the two extremal partitions, those that "stick out most": each of them contains one point of the pair of points that maximizes d on $\mathcal{G}$. Now consider an abstracted graph $\mathcal{T}_1^*$ that consists of the selected extremal partition of $\mathcal{G}$, node ε_1, and a second node u_1 that corresponds to the union of all other partitions of $\mathcal{G}$. We constructed $\mathcal{T}_1^*$ by "cutting out" one partition and "attaching" it to the "rest of the graph".

Consistence of the algorithm. Evidently, $\mathcal{T}_1^*$ is both a tree and reflects the property of $\mathcal{G}$ that an extremal partition corresponds to a leaf in $\mathcal{T}_1^*$. Consider an iterative procedure that conserves these properties of $\mathcal{T}_1^*$: in each iteration, we cut out a new extremal partition from the union of the central remaining partitions. Doing this, the only question is whether one should connect the two new partitions with a new edge or whether one should refrain from it. If one connects the two new

[1] To generate a set of extremal points on the global scale of $\mathcal{G}$, consider the set of points $\mathcal{D}$ that maximize d with respect to each other, that is, they fulfill $x_i = \mathrm{argmax}_{x_j \in \mathcal{G}} \sum_{i' \in \mathcal{D}} d(x_{i'}, x_j) \in \mathcal{D}$. These points span a polygon of most distant points in $\mathcal{G}$. This polygon approximates the topology of the graph only on the largest scale.

bioRxiv preprint first posted online Oct. 25, 2017; doi: http://dx.doi.org/10.1101/208819. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.



Supplemental Figure 11 | Abstracted graph for deep learning based feature space. Analyzing single-cell images via a deep learning based distance metric. Graph abstraction correctly recognizes the cluster of damaged cells as not belonging to the biological path that corresponds to cell cycle evolution through the interphases G1, S and G2. **a,** Abstracted graph with Louvain partitions. **b,** Associated cell cycle phases. **c,** DNA content along a valid path in the abstracted graph. **d,** The DNA content along an invalid path that involves the damaged cells shows a clear non-biological kink.

partitions, this generates a path from the central region of the graph to the extremal partition. If one does not connect the two, one generates two new disconnected leafs, which are connected by the already existing edges. Thereby each step in the iteration identifies a new extremal point and places this extremal in a leaf relative to the remainder of the graph.

Iterating the algorithm. Iterate the following for $i = 2, \dots, n_{\text{partitions}}$.

1. Split u_{i-1} by cutting out a new extremal partition ε_i. The new abstracted graph $\mathcal{T}_i^*$ then has $i+1$ nodes: the new extremal partition ε_i, the new union of the remaining partitions u_i and the $\{\varepsilon_1, \dots, \varepsilon_{i-1}\}$ partitions that already existed in $\mathcal{T}_{i-1}^*$.
2. Attach all edges that already existed and linked u_{i-1} in $\mathcal{T}_{i-1}^*$ to either ε_i or u_i in $\mathcal{T}_i$; among ε_i or u_i, choose the one that is more strongly attached.
3. Decide whether to generate a new edge between ε_i and u_i in $\mathcal{T}_i$. If among all partitions that were not considered in step 2, ε_i and u_i are mutually most strongly attached to each other,
    a) assume that they form a continuous structure that links ε_i to u_i and we add an edge between them, provided this does not generate a cycle in $\mathcal{T}_i$,
    b) otherwise, assume they correspond to disconnected clusters and do not add a new edge.

bioRxiv preprint first posted online Oct. 25, 2017; doi: http://dx.doi.org/10.1101/208819. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.

Notes. One can view the algorithm as iteratively generating partitions by grouping those points that are closer to one extremal point than to the other, with this extremal point; and the rest of the points with the second extremal point. If one does not have a given partitioning of $\mathcal{G}$, one would naturally use the distance measure d to decide which point is closer and hence generate partitions similar to divisive hierarchical clustering on graphs [39]. If one has a given partitioning, one naturally uses this partitioning to group points with extremal points.

Supplemental Note 9: Graph abstraction for datasets of PBMCs

We analyze two PBMC datasets that contain mainly differentiated cells (Supplemental Figure 10). Hence, the abstracted graphs only recover motifs of the PBMC lineage tree and predict many low-confidence connections. See the main text for a discussion.

Supplemental Note 10: Graph abstraction for deep learning

Without extensive preprocessing, the graph of neighborhood relations of data points in gene expression space is useless if computed with a simple fixed distance metric (euclidian, cosine, correlation-based, etc.). If one considers the pixel space of images the problem is even worse and it is impossible to come up with preprocessing methods that lead to a meaningful distance metric. It has recently been shown that a deep learning model can generate a feature space in which distances reflect the continuous progression of cell cycle and a disease [29], that is, deep learning can generate a feature space in which data points are positioned according to biological similarity and by that generates a distance metric that is much more valuable than a simple fixed distance metric. We demonstrate that graph abstraction is useful for reconstructing the cell cycle from image data while and identifying a cluster of damaged cells (Supplementary Figure 11).

References

[1] Wagner, A., Regev, A. & Yosef, N. Revealing the vectors of cellular identity with single-cell genomics. *Nature Biotechnology* **34**, 1145–1160 (2016).

[2] Trapnell, C. *et al.* The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature Biotechnology* **32**, 381–386 (2014).

[3] Bendall, S. C. *et al.* Single-Cell Trajectory Detection Uncovers Progression and Regulatory Coordination in Human B Cell Development. *Cell* **157**, 714–725 (2014).

[4] Note (1). A set of data points can never be an "algebraic manifold" as it lacks any smoothness property; the name "variety" would be a better description. Still the notion "manifold" is established as one often thinks of data as arising from a noisy measurement of the smooth manifold of a dynamical system.

[5] Note (2). The term "graph abstraction" originates from the class of "pattern-based" graph abstraction methods [50, 51]. Their idea is to compute a simple abstraction of a complicated graph based on a set of fixed rules, for example, the contraction of a chain of edges to a single edge in the simple, abstracted graph — similar as in graph coarsening. As applying these exact-rule based algorithms to single-cell data is impractical, confusion with approximate graph abstraction (AGA) is unlikely and we will often just use the notion "graph abstraction".

[6] Qiu, X. *et al.* Single-cell mRNA quantification and differential analysis with Census. *Nature Methods* **14**, 309 – 315 (2017).

bioRxiv preprint first posted online Oct. 25, 2017; doi: http://dx.doi.org/10.1101/208819. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.

[7] Setty, M. *et al.* Wishbone identifies bifurcating developmental trajectories from single-cell data. *Nature Biotechnology* **34**, 637–645 (2016).

[8] Haghverdi, L., Büttner, M., Wolf, F. A., Buettner, F. & Theis, F. J. Diffusion pseudotime robustly reconstructs branching cellular lineages. *Nature Methods* **13**, 845–848 (2016).

[9] Street, K. *et al.* Slingshot: Cell lineage and pseudotime inference for single-cell transcriptomics. *bioRxiv* (2017).

[10] Rizvi, A. H. *et al.* Single-cell topological RNA-seq analysis reveals insights into cellular differentiation and development. *Nature Biotechnology* **35**, 551–560 (2017).

[11] Qiu, P. *et al.* Extracting a cellular hierarchy from high-dimensional cytometry data with SPADE. *Nature Biotechnology* **29**, 886–891 (2011).

[12] Giecold, G., Marco, E., Garcia, S. P., Trippa, L. & Yuan, G.-C. Robust lineage reconstruction from high-dimensional single-cell data. *Nucleic acids research* **44**, e122–e122 (2016).

[13] Grün, D. *et al.* De novo prediction of stem cell identity using single-cell transcriptome data. *Cell Stem Cell* **19**, 266–277 (2016).

[14] Plass, M. *et al.* Cell type atlas and lineage tree reconstruction of whole adult animals by single cell transcriptomics. *submitted* (2017).

[15] Levine, J. H. *et al.* Data-Driven Phenotypic Dissection of AML Reveals Progenitor–like Cells that Correlate with Prognosis. *Cell* **162**, 184–197 (2015).

[16] Weinreb, C., Wolock, S. & Klein, A. SPRING: a kinetic interface for visualizing high dimensional single-cell expression data. *bioRxiv* (2017).

[17] Xu, C. & Su, Z. Identification of cell types from single-cell transcriptomes using a novel clustering method. *Bioinformatics* **31**, 1974–1980 (2015).

[18] Paul, F. *et al.* Transcriptional Heterogeneity and Lineage Commitment in Myeloid Progenitors. *Cell* **163**, 1663–1677 (2015).

[19] Nestorowa, S. *et al.* A single-cell resolution map of mouse hematopoietic stem and progenitor cell differentiation. *Blood* **128**, e20–e31 (2016).

[20] Satija, R., Farrell, J. A., Gennert, D., Schier, A. F. & Regev, A. Spatial reconstruction of single-cell gene expression data. *Nature Biotechnology* **33**, 495–502 (2015).

[21] Zheng, G. X. Y. *et al.* Massively parallel digital transcriptional profiling of single cells. *Nature Communications* **8**, 14049 (2017).

[22] Wolf, F. A., Angerer, P. & Theis, F. J. Scanpy for analysis of large-scale single-cell gene expression data. *bioRxiv* (2017).

[23] Fruchterman, T. M. J. & Reingold, E. M. Graph drawing by force-directed placement. *Software: Practice and Experience* **21**, 1129–1164 (1991).

[24] Csardi, G. & Nepusz, T. The igraph software package for complex network research. *InterJournal Complex Systems* **2006**, 1695 (2006).

[25] Görgens, A. *et al.* Multipotent hematopoietic progenitors divide asymmetrically to create progenitors of the lymphomyeloid and erythromyeloid lineages. *Stem cell reports* **3**, 1058–1072 (2014).

[26] Velten, L. *et al.* Human haematopoietic stem cell lineage commitment is a continuous process. *Nature Cell Biology* **19**, 271–281 (2017).

bioRxiv preprint first posted online Oct. 25, 2017; doi: http://dx.doi.org/10.1101/208819. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.

[27] Stegle, O., Teichmann, S. A. & Marioni, J. C. Computational and analytical challenges in single-cell transcriptomics. *Nature Reviews Genetics* **16**, 133–145 (2015).

[28] Regev, A. *et al.* The Human Cell Atlas. *bioRxiv* (2017).

[29] Eulenberg, P. *et al.* Reconstructing cell cycle and disease progression using deep learning. *Nature communications* **8**, 463 (2017).

[30] Singh, G., Mémoli, F. & Carlsson, G. E. Topological methods for the analysis of high dimensional data sets and 3d object recognition. In *Eurographics Symposium on Point-Based Graphics* (2007).

[31] Mao, Q., Wang, L., Tsang, I. & Sun, Y. Principal Graph and Structure Learning Based on Reversed Graph Embedding. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PP**, 1–1 (2017).

[32] Ji, Z. & Ji, H. TSCAN: Pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis. *Nucleic acids research* **44**, e117 (2016).

[33] Chen, J., Schlitzer, A., Chakarov, S., Ginhoux, F. & Poidinger, M. Mpath maps multi-branching single-cell trajectories revealing progenitor cell progression during development. *Nature Communications* **7**, 11988 (2016).

[34] Qiu, X. *et al.* Reversed graph embedding resolves complex single-cell trajectories. *Nature methods* (2017).

[35] Lovász, L. Random Walks on Graphs: A Survey. *Combinatorics, Paul Erdös is Eighty* **2**, 1 (1993).

[36] von Luxburg, U. A Tutorial on Spectral Clustering. *Statistics and Computing* **17**, 395 (2007).

[37] Coifman, R. R. *et al.* Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the National Academy of Sciences* **102**, 7426–7431 (2005).

[38] Safro, I., Sanders, P. & Schulz, C. Advanced coarsening schemes for graph partitioning (2012).

[39] Pons, P. & Latapy, M. Computing communities in large networks using random walks. *Computer and Information Sciences - ISCIS* 284 (2005).

[40] Fouss, F., Pirotte, A., Renders, J.-M. & Saerens, M. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on knowledge and data engineering* **19**, 355–369 (2007).

[41] Blondel, V. D., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech.* **2008**, P10008 (2008).

[42] Newman, M. E. J. & Girvan, M. Finding and evaluating community structure in networks. *Phys. Rev. E* **69**, 026113 (2004).

[43] van der Maaten, L. & Hinton, G. Visualizing Data using t-SNE. *Journal of Machine Learning Research* **9**, 2579–2605 (2008).

[44] Amir, E.-a. D. *et al.* viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia. *Nature Biotechnology* **31**, 545–552 (2013).

[45] Wang, B., Zhu, J., Pierson, E., Ramazzotti, D. & Batzoglou, S. Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning. *Nature methods* **14**, 414–416 (2017).

bioRxiv preprint first posted online Oct. 25, 2017; doi: http://dx.doi.org/10.1101/208819. The copyright holder for this preprint (which was not peer-reviewed) is the author/funder. It is made available under a CC-BY-NC-ND 4.0 International license.

[46] Traag, V. Louvain. *GitHub* (2017).

[47] Krumsiek, J., Marr, C., Schroeder, T. & Theis, F. J. Hierarchical Differentiation of Myeloid Progenitors Is Encoded in the Transcription Factor Network. *PLoS ONE* **6**, e22649 (2011).

[48] Moignard, V. *et al.* Decoding the regulatory network of early blood development from single-cell gene expression measurements. *Nature Biotechnology* **33**, 269–276 (2015).

[49] Wittmann, D. M. *et al.* Transforming Boolean models to continuous models: methodology and application to T-cell receptor signaling. *BMC Syst. Biol.* **3**, 98 (2009).

[50] Boneva, I., Rensink, A., Kurban, M. & Bauer, J. Graph Abstraction and Abstract Graph Transformation. Tech. Rep., Centre for Telematics and Information Technology, University of Twente, Enschede (2007).

[51] Rensink, A. & Zambon, E. *Pattern-Based Graph Abstraction*, 66–80 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2012).