Joint analysis of heterogeneous single-cell RNA-seq dataset collections

Nikolas Barkas1,4, Viktor Petukhov1,2,4, Daria Nikolaeva1, Yaroslav Lozinsky1, Samuel Demharter2, Konstantin Khodosevich2 and Peter V. Kharchenko1,3*

Single-cell RNA sequencing is often applied in study designs that include multiple individuals, conditions or tissues. To identify recurrent cell subpopulations in such heterogeneous collections, we developed Conos, an approach that relies on multiple plausible inter-sample mappings to construct a global graph connecting all measured cells. The graph enables identification of recurrent cell clusters and propagation of information between datasets in multi-sample or atlas-scale collections.

The progress of single-cell RNA sequencing (scRNA-seq) techniques has enabled individual groups to measure dozens of samples, often in complex designs incorporating treatment and control cases, disease and normal pathology, or multiple tissues. Consortium efforts are under way to generate atlases of single-cell datasets covering diverse biological contexts with thousands of samples1–3. Joint consideration of such panels poses technical and conceptual analysis challenges, necessitating new methods and reconsideration of the aims. In contrast to the traditional batch correction problem, where inter-sample variation can be treated as a technical artifact that needs to be controlled for4–5, the panels can include systematically different samples, with some of the datasets lacking any shared cell subpopulations. Recent alignment methods6–8, while significantly more flexible, were designed to align relatively small sample panels with modest compositional variation. We therefore set out to develop an approach for analyzing and navigating large heterogeneous sample collections.

We reasoned that a unified graph representation could capture likely relationships between cells in different samples, and that statistical analysis of such a joint graph can identify subpopulations across different samples (Fig. 1a). To construct the joint graph, Conos (clustering on network of samples) performs pairwise alignments of individual designs, identifying plausible inter-sample cell–cell correspondence (inter-sample edges). Such mappings are error prone, as inter-sample variation cannot be usually modeled or constrained. Across many pairwise comparisons, however, the recurrent subpopulations of cells will tend to map to each other, forming clique-like communities within the joint graph that can be identified over the background noise of spurious edges (Fig. 1a). Conos also adds low-weight edges connecting neighboring cells within the individual samples, as a weak prior for preserving local neighborhoods of cells in each sample. The plausible mapping between a pair of samples is established using mutual nearest-neighbor (mNN) mapping in reduced expression space6,7. We evaluated spaces capturing common variation across two or more samples, including common principal component analysis8 (CPCA), joint non-negative matrix factorization and higher-order generalized singular value decomposition9.

We first applied Conos to a collection of 16 scRNA-seq samples of human bone marrow (BM) and cord blood from the Human Cell Atlas5. Projection of the resulting joint graph separated all major subpopulations (Fig. 1b and Supplementary Fig. 1), with the detected joint cell clusters connecting the corresponding subpopulations across the entire collection. While the individual samples were well mixed within the joint graph, the systematic difference in the composition of the two tissues was also apparent (Fig. 1c). To quantify robustness and sensitivity, we perturbed the full dataset to decrease signal or increase heterogeneity between samples. Examining recovery of the original subpopulations identified by each method under decreasing numbers of cells (Fig. 1d) or under decreasing magnitude of subpopulation-specific expression signatures (Supplementary Fig. 1e), we found that Conos significantly outperformed earlier methods. Conos performance remained robust under parameter perturbations (Supplementary Note 1), and even with simple pairwise alignment strategies (for example, nearest-neighbor mapping based on simple gene correlation; see Supplementary Fig. 1). We have applied Conos to reanalyze a number of recently published complex datasets7–12, in all cases joining corresponding annotated subpopulations across different samples and tissues (Supplementary Figs. 2–7).

Modern experimental designs are likely to combine distinct classes of samples within the panels, such as sets of disease samples and healthy controls, or multiple tissues from different individuals. We simulated increasing heterogeneity of a panel by omitting increasing numbers of random clusters from samples and evaluating the method’s ability to recover originally detected subpopulations (Fig. 1e). Conos showed higher robustness compared to other methods. Furthermore, Conos was able to sustain uniform mixing (that is, high entropy) of cells from different samples among the identified joint clusters even under high sample heterogeneity, where some of the samples shared few or no cell subpopulations (Fig. 1f,g). Community detection on a joint graph shows high sensitivity, for instance, enabling Conos to detect subpopulations that may be represented by only a single cell in a given sample (Supplementary Fig. 1f). More important, the sensitivity of the proposed approach increased as more samples were added to the panel (Fig. 1h), indicating that larger collections of scRNA-seq samples will reveal more subtle recurrent cell subpopulations.

Consideration of diverse sample panels requires one to re-examine the aims of integration. While for homogeneous panels, the aim is to identify a set of clusters that appear in nearly every
sample, this is not the case for panels with distinct classes of samples. For example, for a panel containing both tumor and adjacent normal samples\(^\text{12}\), it would not be desirable to lump tumor cells with any normal tissue subpopulations, even though the cluster of tumor cells would be restricted to a subset of samples (for example, Supplementary Fig. 5). In a more nuanced scenario, the clustering may separate tumor-associated CD4\(^+\) T cells from their counterparts in the healthy tissues\(^\text{9}\), picking up persistent biological difference in their state (Fig. 2a–d). However, for annotating major cell types, a unified cluster of CD4\(^+\) T cells across all samples would be more
appropriate. Graph communities can be viewed as a hierarchical clustering of cells, and in that way the difference between separating or joining tissue-specific subgroups of CD4⁺ T cells is equivalent to cutting the hierarchy at different levels (Fig. 2d,e and Supplementary Fig. 5). Overall, lower cuts will yield higher resolution of subpopulations, but will also decrease cluster breadth—the average fraction of samples in which a cluster appears. As the balance between the desired resolution and breadth will depend on the question being posed by the investigator, Conos incorporates an interactive tool to explore the hierarchical community structures (Supplementary Fig. 8). For the situations when a higher degree of mixing between samples is desired, Conos implements an option to increase ‘alignment strength’ by sampling cell–cell edges from neighborhoods of higher radii and rebalancing edge weights (see Methods). Adjustment of this continuous ‘alignment strength’ parameter and the optional edge weight rebalancing based on the sample type enables higher mixing at the expense of resolution (Supplementary Fig. 9).

A joint graph can be used to map properties between samples by simulating the diffusion process. For instance, one can propagate discrete cell annotation labels to datasets that have not yet been annotated (Fig. 2f). On the BM example, Conos propagated labels from one dataset to the other seven with 97% accuracy (Fig. 2g). The diffusion propagation keeps track of uncertainty, and almost all of the misclassified cells were reported to have high uncertainty of the labels (Fig. 2h,i). Similarly, diffusion of gene expression magnitudes provides a way of deriving common expression space (Supplementary Fig. 10). Such ‘corrected’ expression values are estimated by most of the existing batch correction or alignment methods. We contend that the use of such corrected expression values will be mostly limited to visualization. Once the appropriate clustering of cells is established, we expect follow-up analyses to focus on the expression variation among samples. This includes cell-type-specific analysis of expression differences between groups of samples, or variation within groups. Corrected expression values specifically attenuate differences between samples, and would lead to inflated significance estimates in comparisons of different cell subpopulations. Instead, Conos reformulates the differential expression tests as comparisons of in silico bulk RNA-seq measurements that can be delegated to common differential expression software, providing convenience routines for the common differential tests.

Overall, our results demonstrate that integration of single-cell collections into a unified graph representation provides effective means for joint analysis, including subpopulation identification, differential expression or annotation. Compared to current methods, Conos implementation shows improved stability and sensitivity, particularly notable on heterogeneous sample panels, such as multi-tissue/multi-patient clinical study designs. This robustness allows Conos to wire together very diverse collections of samples, such as organ-scale atlases where most of the samples will have few or no cell types in common. For example, we reanalyzed the Tabula Muris mouse atlas, combining 48 datasets covering different mouse tissues (Supplementary Figs. 11–13), and further combined it with another atlas by Han et al. The resulting joint graph integrated a total of 127 individual datasets, containing 419,405 cells, and was effective at identifying common cell populations across samples measuring diverse tissues, as well as overcoming the differences of the three different scRNA-seq platforms used (Supplementary Figs. 14–18). The approach is fast, particularly when using a simpler principal component analysis space (Supplementary Fig. 19). Conos can also be applied to other molecular modalities. As an example, we used Conos to assemble a mouse single-cell chromatin accessibility atlas, as well as to integrate accessibility and scRNA-seq data (Supplementary Note 2).
We hope that the presented approach will enable other research groups to effectively interpret scRNA-seq collections in complex experimental designs.

Online content
Any methods, additional references, Nature Research reporting summaries, source data, statements of code and data availability and associated accession codes are available at https://doi.org/10.1038/s41592-019-0466-z.

Received: 5 December 2018; Accepted: 24 May 2019; Published online: 15 July 2019

References
1. Tabula Muris Consortium. Nature 562, 367–372 (2018).
2. Regev, A. et al. elife 6, e27041 (2017).
3. Hicks, S. C., Townes, F. W., Teng, M. & Irizarry, R. A. Biostatistics 19, 562–578 (2018).
4. McCarthy, D. J., Campbell, K. R., Lun, A. T. & Wills, Q. F. Bioinformatics 33, 1179–1186 (2017).
5. Butler, A., Hoffman, P., Smibert, P., Papalexi, E. & Satija, R. Nat. Biotechnol. 36, 411–420 (2018).
6. Haghverdi, L., Lun, A. T. L., Morgan, M. D. & Marioni, J. C. Nat. Biotechnol. 36, 421–427 (2018).
7. Neuenschwander, B. E. & Flury, B. D. J. Multivar. Anal. 75, 163–183 (2000).
8. Ponnappalli, S. P., Saunders, M. A., Van Loan, C. F. & Alter, O. PLoS ONE 6, e28072 (2011).
9. Azizi, E. et al. Cell 174, 1293–1308 e1236 (2018).
10. Puram, S. V. et al. Cancer Cell 171, 1611–1624 (2017).
11. Guo, X. et al. Nat. Med. 24, 978–985 (2018).
12. Lambrechts, D. et al. Nat. Med. 24, 1277–1289 (2018).
13. Lun, A. T. L. & Marioni, J. C. Biostatistics 18, 451–464 (2017).
14. Love, M. I., Huber, W. & Anders, S. Genome Biol. 15, 550 (2014).
15. Ritchie, M. E. et al. Nucleic Acids Res. 43, e47 (2015).
16. Han, X. et al. Cell 172, 1091–1107 (2018).
17. Cusanovich, D. A. et al. Cell 174, 1309–1324 (2018).
18. Cao, J. et al. Science 361, 1380–1385 (2018).

Acknowledgements
N.B. and P.V.K. were supported by the NIH R01HL131768 and NSF-14-532 CAREER awards. D.N. and Y.Z. were supported by the SMTB Alumni Summer Research Program from Zimin Foundation. We would like to thank the HMS Research Computing team for facilitating benchmarking calculations using the O2 cluster.

Author contributions
N.B., V.P. and P.V.K. implemented the method and ran evaluations. D.N. evaluated label transfer benchmarks and helped with implementation. Y.L. implemented the interactive view of hierarchical communities. S.D. and K.K. applied the method to integration of neuronal atlases. P.V.K. designed and oversaw the study and drafted the manuscript with help from V.P.

Competing interests
P.V.K. serves on the Scientific Advisory Board to Celsius Therapeutics Inc.

Additional information
Supplementary information is available for this paper at https://doi.org/10.1038/s41592-019-0466-z.

Reprints and permissions information is available at www.nature.com/reprints.

Correspondence and requests for materials should be addressed to P.V.K.

Peer Review Information: Nicole Rusk was the primary editor on this article and managed its editorial process and peer review in collaboration with the rest of the editorial team.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.
© The Author(s), under exclusive licence to Springer Nature America, Inc. 2019
Methods

Overview of the approach. Conos processing can be divided into several key phases. During phase I, each individual dataset in the sample panel is filtered and normalized using standard packages for single-cell-processing: pagoda2 or Seurat. Specifically, Conos relies on these methods to perform cell clustering, library size normalization, identification of overdispersed genes, and, in the case of pagoda2, variance normalization. Conos is robust to variations in the normalization procedures, but it is recommended that all of the datasets be processed uniformly. During phase II, Conos performs pairwise comparisons of the datasets to establish initial, error-prone, mappings between cells of different datasets. These inter-sample edges are then combined with lower-weight intra-sample edges during phase III—joint graph construction. The joint graph is then used for downstream analysis, including community detection, label propagation and so on.

Pairwise dataset alignments (phase II). Initial inter-sample edges between a given pair of datasets i and j were established based on a choice of (1) rotation space and (2) neighbor mapping strategy (nearest neighbor or mNN). For each dataset, a set of overdispersed (hypervariable) genes (g, p) was determined using pagoda2 (default n.osgenes = 2,000 top overdispersed genes were used), and an union of overdispersed genes from both datasets was taken, limiting the genes to those for which the data were available in both datasets: g = (g ∪ U g) ∩ G ∩ G', where G and G' are the full gene sets for the two datasets. Subsequent analysis was carried out on matrices M and M', with columns corresponding to genes g, and rows corresponding to the cells in each dataset. The entries of each matrix were taken to be the variance-normalized expression magnitudes determined by pagoda2 (normalized expression magnitudes are used if pre-processing was performed by Seurat).

The reduced projection matrices R and R' were obtained according to the space used: for CPCA and JNMF (joint non-negative matrix factorization), these corresponded to projections onto the corresponding common/joint components (30 components, by default). For PCA space, principal components (30 by default) were determined independently for M and M', with R and R' then determined by projecting the cells of each dataset onto a joint (that is, 60-dimensional) space of both sets of principal components. For gene space, R and R' were then taken to be the matrices M and M' themselves.

Cell–cell similarity between cells k and l was determined as wk,l = max(0, rkl), where rkl is the Pearson linear correlation between the ith row of M and the jth row of M'. An alternative L2 distance was implemented as wk,l = exp(−||μk − μl||2/α), where the default scaling constant α = 10.

Joint graph (phase III). For a given sample collection, the nodes of the joint graph G correspond to all of the cells included in the collection, connected by a combination of inter- and intra-sample edges. The inter-sample edges were determined as mNN (default) or plain nearest neighbors, with the weight wk,l. Neighborhood size k = 15 was used by default. Intra-sample edges for each dataset i connected each cell to kadj (default kadj = 5) cells within the dataset i using weights wk,l = ci,adj Xi Xj within the reduced space R as determined by the projection of the cells onto the top principal components of M. The constant ci,adj = 0.1 was used to reduce the contribution of the intra-sample edges relative to the inter-sample edges.

Joint clustering. Joint clusters were determined as communities of the joint graph G, using standard community detection methods. By default, the walktrap communities algorithm implemented by the igraph package was used, with step = 20. Louvain clustering implemented via the igraph:multilevel communities method provided much faster performance, but it lacked hierarchical output. Implementation of the Leiden community detection method with the resolution parameter was adapted from https://github.com/vtraag/leidenalg.

Alignment strength. By default, Conos aims to preserve biological variation by keeping track of cell–cell distances using edge weights and treating all comparisons in a symmetric way. In some cases, however, the user may want to force a greater degree of alignment (mixing) of the datasets. This results in a trade-off between resolution (ability to resolve a finer subpopulation) and mixing of samples (Supplementary Fig. 8). To provide such a control, we added the α parameter that allows an increase of the nearest-neighbor search radii. k, is initially used instead of k during mNN-graph construction. Then, the edges are pared down to reduce the maximal degree of the graph vertices close to k, making the graph less dense and more regular. The following greedy procedure is used:

1. Vertices are ordered from the highest to the lowest degree.
2. For each vertex, edges are ordered by the degree of their target vertices (high to low).
3. The algorithm iterates through the vertices and corresponding edges, removing an edge if the degrees of both incident vertices are larger than a specific cut-off kα.

As this is a greedy algorithm, to achieve more uniform reduction, Conos runs the procedure iteratively, reducing k, from k, to k, using a logarithmic grid of three steps. We also observed that even with very large k, (when trying to force alignment of very distant cells), Conos relies on these methods to perform cell clustering. In other cases, positive correlation rkl. To prevent that, the weight calculation can be changed from wk,l = max(rkl, 0) to wk,l = 1 + rkl (using cor.base = 2 argument; by default the cor.base value will be increased toward two automatically when increasing k,). To provide a user-friendly way to control k, we also implemented an alignment strength parameter α (alignment strength), such that k, = α k, where k, is the maximal number of total cells among the samples in the panel. Thus, α varies from 0 to 1, where 0 (default) corresponds to the alignment with no additional edges and 1 corresponds to a full (non-informative) alignment with uniform edge placement over the graph.

Rebalancing of edge weights. In many experimental designs, samples can be classified by an extraneous factor, such as subject group (for example, healthy versus disease) or protocol (for example, Smart-seq2 versus 10x Chromium). As discussed previously in the article, such systematic differences should typically lead to hierarchically defined factor-specific subclusters within each major subpopulation. However, in some cases, the user may want to explicitly force alignment across such factors. To implement such control, we added an optional step that balances weights of the edges connecting cells from samples belonging to the same or different values of the factor. This is achieved by modifying the following function:

$$\sum_{i=1}^{N_{samples}} \sum_{j=1}^{N_{adjacent}} \sum_{k=1}^{N_{factors}} \left| \frac{w_{i,j} - w_{adj(i),adj(j)}}{w_{adj(i),adj(j)}} \right|$$

where Nfactors is the total number of factor levels; Nsamples is the total number of cells in the dataset; adj(i) is the set of cells adjacent to the cell i; adj(j) is the set of cells adjacent to j and belonging to the factor level l; w_{i,j} is the weight of the edge between cells i and j; and N_{factors} is the number of different factors among cells connected to s. The minimization is performed using a two-step procedure. The first step estimates the imbalance ratio for a cell i and a factor level l: \(\mu_{i,l} = \frac{\sum_{s \in \text{factor level } l} w_{i,s} \text{adj(}s\text{)} - w_{i,i}}{\sum_{s \in \text{factor level } l} w_{i,s}}\). The second step updates the edge weights as \(w_{i,j} = w_{i,j} + \mu_{i,l} \cdot w_{adj(i),adj(j)}\), where l denotes the factor level of the cell. This procedure is repeated 50 times, which does not guarantee convergence but allows for reduction of the loss function by several orders of magnitude in a reasonable time. Such optimization preserves the ratios of the edge weights to l, varying only the weight ratios between the edges connecting cells with different factor levels. Further prioritization of the edges connecting samples from different factor levels can be gained using the same factor. The final weight parameter (which, when set below the default value of 1, will reduce the weight of edges connecting cells of the same factor level). One can also apply an edge rebalancing procedure without an extraneous grouping of samples by assigning each sample to its own factor level.

Label and value propagation. Propagation of both labels and expression magnitudes was treated as a general problem of information propagation between graph vertices. Graph vertices can have multiple labels, either continuous or discrete. For visualization purposes, joint graphs were laid out in two dimensions using an iterative diffusion process on the joint graph. For continuous labels the diffusion process was implemented as follows:

1. Vertices are ordered from the highest to the lowest degree.
2. For each vertex, edges are ordered by the degree of their target vertices (high to low).
3. The algorithm iterates through the vertices and corresponding edges, removing an edge if the degrees of both incident vertices are larger than a specific cut-off k,.

Considering each gene as a continuous label, Conos uses this diffusion process to correct gene expression matrices and bring all of the datasets into a ‘common’ expression space. We note that a single iteration of the diffusion process with parameter \(\alpha = 0\) is equivalent summing of expression over adjacent cells, which is a common approach for noise correction in scRNA-seq data.

For discrete labels, the implementation tracked label uncertainty, with the diffusion process being used to estimate the posterior probability of each label for each vertex. We did this by running diffusions on the probability distribution of the labels:

1. The posterior distribution of possible labels was kept for each vertex. The starting state for the labeled vertices was set so that the probability of the
true label was set to 1, with the probability of other values set to 0. For the unlabeled vertices, all of the values were initially set to 0.

2. On these distributions we simultaneously simulated the diffusion process for each component (or distribution (that is, each label)). After each diffusion step, the values of the posterior distributions were re-normalized so that the sum of the label probabilities was equal to 1.

Diffusion of discrete values was used for the cell annotation propagation results (Fig. 2f–h). In the figures, the uncertainty of the labeling was evaluated as $1 - \max(p_i)$.

**Benchmark design.** Quantitative performance of different methods shown in Fig. 1 and Supplementary Fig. 1 relied on the same general design, where each method $m$ was run on a full dataset to obtain clustering $C_m$. The full dataset was then gradually perturbed to pose a more challenging problem, and the ability of different methods to recover their corresponding original clustering $C_0$ was measured. Such a procedure aimed to place different methods on even footing and make use of realistic data (as opposed to synthetic). Details of different benchmarks are given next.

For the cell subsampling benchmark (Fig. 1a), a Human Cell Atlas bone marrow plus cord blood (HCA BM + CB) $3k$ dataset containing a total of 16 samples was used (see below for dataset details). A percentage of $p_{\text{perturb}} \approx 50\%$ was randomly sampled and removed from the full dataset. For each value of $p_{\text{perturb}}$, each of ten replicates of dataset perturbation were generated. To assess performance, the adjusted Rand index $(\gamma)$ was calculated relative to the first replicate with $p_{\text{perturb}}=0$. A smoothed mean for each method and the corresponding 95% confidence band is shown in Fig. 1c, which were calculated using the igraph::geom_smooth() method. Note that all of the examined methods showed a certain level of instability when faced with negligible perturbations of the dataset (such as shuffling of the cell order in the dataset, or shuffling one cell in a single cell). As datasets sampled with $p_{\text{perturb}}=0$ shuffled the order of the cells, the adjusted Rand index value at $p_{\text{perturb}}=0$ was below 1.

For the cell mixing benchmark (Supplementary Fig. 1c), a HCA BM + CB 3k dataset was used. For each dataset $i$, a background expression vector $\mathbf{b}$ was determined for each gene $g$ as $\mathbf{b}_g = N_i^g$, where $N_i^g$ is the total number of molecules of gene $g$ detected in the dataset $i$. A perturbed dataset with a mixing proportion $p_{\text{mix}} \in [0, 1]$ was generated for each cell $i$ by iterating through each molecule of the cell, keeping the original molecule with a probability $1 - p_{\text{mix}}$, or alternatively (with probability $p_{\text{mix}}$) replacing it with a molecule randomly sampled from the background profile, $\mathbf{b}$. This way, datasets generated with $p_{\text{mix}} = 0$ were equivalent to the original data, whereas $p_{\text{mix}} = 1$ yielded datasets where each cell was a random sampling of the background and any cell subpopulations would be impossible to discern.

For the cluster dropping benchmark (Fig. 1f–h), to simulate increasing compositional variability between the samples, cells belonging to a cluster $c \in C_0$ were omitted with a probability $p_{\text{omit}}$. The sampling procedure was carried out independently for each dataset $i$, with $10\%$ of cells randomly dropped from different datasets (increasing compositional variability). To guarantee a minimal dataset size, a total of five clusters were sampled this way in each dataset. Under such a procedure, $p_{\text{omit}}=0$ maintains the full original dataset, whereas $p_{\text{omit}}=1$ maximizes inter-sample compositional differences.

Tests to the cells from the different samples were mixed within the resulting clusters was quantified using normalized relative entropy, weighted by the cluster size

$$\sum_{k=1}^{n_{\text{clusters}}} \frac{\text{KL}(k(\mathbf{f}, F))}{\text{KL}(\mathbf{f}, F)} \log \left( \frac{n_{\text{clusters}}(k)}{n_{\text{clusters}}(k) \sum_{k=1}^{n_{\text{clusters}}} n_{\text{clusters}}(k)} \right)$$

where $k_i$ is a vector giving the number of cells from each sample in a cluster $k$, $\text{KL}(\mathbf{f}, F)$ is the empirical KL divergence (relative entropy) between the $k_i$ and the total number of cells in each dataset $\mathbf{F}$ (calculated using KL empirical from the entropy package in R), $s_i$ is the total number of cells in a cluster $k$, $n_{\text{clusters}}$ is the number of clusters detected by the method on a current realization of the dataset and $n_{\text{clusters}}$ is the total number of samples in the panel. As we expected to observe systematic composition differences between BM and cord blood, the normalized entropy was assessed separately for BM and cord blood cells (Fig. 1g,h, respectively).

For the number of stable clusters (Fig. 1i), to assess how the number of stable clusters changes with increasing size of the sample panel, we assembled a larger panel of samples covering the same tissue (HCA BM + 10x RB dataset). Ten randomized ‘series’ were constructed, with each series starting with two randomly chosen datasets and then adding one dataset per step up to a maximum of ten available datasets (sampling without replacement was used to construct the series). As community detection algorithms rely on heuristics such as maximizing of modularity, we evaluated the number of stably detectable clusters as a number of independent subtrees in the hierarchy returned by the walktrap.community algorithm. A stable subtree was determined as a subtree containing at least 30 cells that can be detected under a 10% cell subsampling perturbation (see below) with the Jaccard coefficient to the best matching subtree above 0.8. To evaluate these stability properties, for each run an additional ten subsampling runs were made, omitting 10% of the cells of the sample and rerunning walktrap.community to generate the perturbed trees on the basis of which the Jaccard coefficient was calculated.

For dN/dS analysis with individual samples (Supplementary Fig. 1j), to evaluate how well different methods were able to pick up rare cells in the dataset, we simulated rare cell occurrences by randomly choosing a single sample in the panel, choosing a random joint cluster $c \in C_0$ that occurs within that dataset, and then leaving only one randomly selected cell from that cluster $c$ within that sample. The HCA BM + CB $1k$ panel was used containing 16 samples. In total, $16 \times 5 \times 5 = 400$ perturbed panels were generated, sampling five different clusters $c$ from each of the 16 datasets, with five different random choices of the retained cell being made. In evaluating the performance, the remaining cell from the cluster $c$ was scored as correctly classified if it was assigned to a cluster to which other cells of a cluster $c$ were most commonly assigned.

For control of alignment strength (Supplementary Fig. 9), to evaluate the effectiveness of the optional alignment strength and edge weight rebalancing parameters, we used an example of the human pancreas islet datasets from 10x Chromium, inDrop and Smart-Seq2 protocols. Individual samples were separated according to the provided annotations, and normalized using Pagoda2 with parameters $\text{npOAs} = 100$ and $\text{n.ogd} = 1,000$. Pagoda2 objects were aligned using Conos with PCA space ($k = 30$, $k.self=5$, cor.base = 2). For each value of the alignment strength parameter $(\sigma \in [0.0, 0.1, 0.2, 0.3, 0.5, 1.0])$, the analyses were run with and without edge weight balancing. Because cell-type annotation for the Smart-Seq2 protocol was not available, the Conos label propagation procedure was used to label these cells (Supplementary Fig. 9a: label transfer was performed using alignment generated using default Conos parameters; that is, $\sigma = 0.0$ and no edge weight balancing).

For each combination of the alignment strength and edge balancing parameters, we obtained a joint graph representing the dataset alignment. For each such graph $G$, we then estimated a value of the resolution parameter $R_i$ of Leiden, which yields approximately 100 clusters of size $\geq 10$ cells. Then, for each $G$, we ran Leiden clustering while varying the resolution parameter on a uniform grid between 0.1 and $R$ with 15 intermediate points. For each of the resulting clusterings, we estimated the normalized relative entropy against the protocol factor (Supplementary Fig. 9b). Finally, as a visual check, we visualized the embeddings of the graphs using largeVis (Supplementary Fig. 9c).

For the assessment of runtime performance, Supplementary Fig. 19 shows runtime central processing unit and memory requirements of Conos under two different scenarios: (1) cell subsampling benchmark detailed above and (2) integration of increasing numbers of datasets. The latter was done by taking an increasing number of random 1k cell draws from the first HCA BM dataset, thus combining up to 100 such simulated datasets. Memory usage was approximated based on the total size of the session (which may not account for transient spikes of memory usage). We note that while the current implementation performs $O(N^3)$ comparisons between samples $N$, the overall runtime, especially in PCA space, is fast enough to allow for integration of hundreds of datasets. Simple schemes to reduce runtime and memory requirements can be devised for very large collections, such as limiting pairwise alignments to a certain random fraction of all possible comparisons. Performance was assessed with a cluster of nodes equipped with Xeon E5-26XX family central processing units, using 12 cores per process.

For implementation of other methods, Conos performance was compared with the two previously published methods and configured in the following way. The Seurat package was initiated from CRAN v.1.3.4. The previous cell annotation and dataset alignment was run as recommended in the tutorial at http://satijalab.org/seurat/immune_alignment.html. The results presented used a default resolution. Using alternative resolution parameters (0.6, 1.4) did not affect the performance significantly.

For enabling certain large-scale benchmarking, we limited the number of common space components estimated by Conos and the two methods above to 20.

**Reporting Summary.** Further information on research design is available in the Nature Research Reporting Summary linked to this article.

**Data availability.** HCA BM and cord blood data were downloaded from the HCA portal (https://preview.data.humancellatlas.org/). The dataset represents a relatively uniform collection of data on well-documented tissues, making it particularly suitable for benchmarking purposes. To reduce calculation times in benchmark evaluations,
we took a random subset of the cells from lane 1 of each dataset. By default, 3,000 cells per sample were used (HCA BM + CB 3k datasets). A smaller, 1,000-cell dataset (HCA BM + CB 1k) was used for the more extensive sensitivity analysis (Supplementary Fig. 1f). For Fig. 1i, we combined HCA BM samples with two samples ('Frozen BMMCs Healthy Donors 1 and 2') downloaded this from 10x Genomics (https://www.10xgenomics.com/resources/datasets/). This was done to extend the number of samples (x axis in Fig. 1i). The data on breast cancer from Azizi et al. were downloaded from GEO (GSE114725) as a count matrix, together with the provided annotations. As shown in the plots (Fig. 2 and Supplementary Fig. 4), the annotations were simplified to collapse patient-specific populations and omit smaller subpopulation distinctions. To demonstrate applicability to different levels of data fragmentation, we reanalyzed the dataset by combining eight individual subjects, 15 subject + tissue combinations or 53 subject + tissue + replicate combinations. The dataset provides a good example of a clinically oriented panel with both tissue- and individual-level heterogeneity. The molecular count data and annotations on lung cancer from Lambrechts et al. were downloaded from ArrayExpress (E-MTAB-6149, E-MTAB-6653). The dataset provides an example of a more typical case-control design of a clinically oriented panel. The molecular count data and annotations on non-small-cell lung cancer from Guo et al. were downloaded from GEO (GSE99254). The dataset serves as an example of a heterogeneous clinically oriented panel, with limited complexity and numbers of cells in some of the samples. The molecular count data and annotations on head-and-neck cancer from Puram et al. were downloaded from GEO (GSE105322). Similar to the data from Guo et al., the dataset provides an example of a collection with challenging complexity and cell-number variation in a clinically oriented panel. For the human cortex comparison, the datasets were included as an example of integration of distinct nuclei-based protocols. The count matrix for Hodge et al. (bioRxiv; https://doi.org/10.1101/384826) was downloaded from http://celltypes.brain-map.org/rnaseq. The count matrix from Lake et al. (Nat. Biotechnol. 36, 70–80; 2018) was downloaded from GEO (GSE97930). Tabula Muris mouse data were downloaded from https://tabula-muris.ds.czbiohub.org/. Only cells with at least 1,000 molecules were analyzed. A total of 48 datasets were combined. The mouse cell atlas by Han et al. and the relevant annotations were downloaded from http://bis.zju.edu.cn/MCA/. Cell line datasets were excluded. Human pancreas islet data from different platforms, used to demonstrate alignment between different platforms and illustrate mixing controls (Supplementary Fig. 9), were taken from the following sources: 10x Chromium platform data were taken from a publication by Xin et al. and downloaded from GEO (GSE114297). Normalized count matrices were used. inDrops platform data were taken from a publication by Baron et al. and downloaded from GEO (GSE84133). Only human data (four samples) were used. Normalized count matrices were used. Smart-seq2 platform data were taken from a publication by Segerstolpe et al. with count matrices downloaded from ArrayExpress (E-MTAB-5061). Only data from healthy individuals (six samples) were used. For the demonstration of ATAC-seq alignment and alignment between ATAC-seq and RNA-seq (Supplementary Note 2), the following datasets were used: sci-ATAC data from Cusanovich et al. were downloaded from the authors’ website (http://atlas.gs.washington.edu/mouse-atac/). Author-provided accessibility scores were used as gene-level input to Conos. sci-CAR data from Cao et al. were downloaded from GEO (GSE117089). To increase coverage, the cells were aggregated into groups of ten on the basis of transcriptional similarity (see Supplementary Note 2 for details).

**Code availability**

Conos is implemented as an R package with C++ optimizations, and is available on GitHub (https://github.com/hms-dbmi/conos) under the GPL-3 open source license. Analysis scripts and intermediate data representations used for the preparation of the manuscript can be found on the author’s website (http://pklab.med.harvard.edu/peterk/conos/).

**References**

19. Xin, Y. et al. *Diabetes* **67**, 1783–1794 (2018).
20. Baron, M. et al. *Cell Syst.* **3**, 346–360 e344 (2016).
21. Segerstolpe, A. et al. *Cell Metab.* **24**, 593–607 (2016).
Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see Authors & Referees and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

- The exact sample size (n) for each experimental group/condition, given as a discrete number and unit of measurement
- A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
- The statistical test(s) used AND whether they are one- or two-sided
- Only common tests should be described solely by name; describe more complex techniques in the Methods section.
- A description of all covariates tested
- A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
- A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
- For null hypothesis testing, the test statistic (e.g. F, t, r) with confidence intervals, effect sizes, degrees of freedom and P value noted
  Give P values as exact values whenever suitable.
- For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
- For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
- Estimates of effect sizes (e.g. Cohen's d, Pearson's r), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection

In demonstrating the developed method we used only published data, describing the relevant publications and where the data was downloaded in the "Data sources and dataset-specific analysis details" subsection of the Methods

Data analysis

The implementation of the developed is available on github (https://github.com/hms-dbmi/conos) under an open-source license (GPL-3).

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors/reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

No original data is included in this manuscript; accession numbers or relevant URLs are provided for all the published datasets used

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

- Life sciences
- Behavioural & social sciences
- Ecological, evolutionary & environmental sciences
### Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

| Sample size | Same size was not determined beforehand. Datasets from published studies were used. |
|-------------|-------------------------------------------------------------------------------------|
| Data exclusions | Cell line datasets included in Han et al. Mouse atlas were excluded, as noted in the Data subsection of the Methods. Criterium (cell line as opposed to primary tissue) was pre-established. |
| Replication | The analysis relied on published datasets (all of which include replicates). No attempt at replicating published studies was done. |
| Randomization | In silico randomizations procedures were used in the benchmarks presented in Figure 1, and are detailed in the Methods. Briefly, the procedures included: (1) random subsampling of a fraction of cells, and (2) random subsampling of a fraction of subpopulations from individual samples. The fractions were varied systematically (shown on x axes). Cluster stability was assessed based on random subsampling of 95% of cells. |
| Blinding | All of the analysis relied on published datasets, so blind designs were not applicable. |

### Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

| Materials & experimental systems | Methods |
|---------------------------------|---------|
| n/a | Involved in the study |
| ☒ | Antibodies |
| ☒ | Eukaryotic cell lines |
| ☒ | Palaeontology |
| ☒ | Animals and other organisms |
| ☒ | Human research participants |
| ☒ | Clinical data |
| n/a | Involved in the study |
| ☒ | ChIP-seq |
| ☒ | Flow cytometry |
| ☒ | MRI-based neuroimaging |