Quantifying distribution shifts in single-cell data with scXMatch

Anna Möller[1,2,*], Miriam Schnitzerlein[5,6,7], Eric Greto[2,3,4], Vasily Zaburdaev[5,6], Stefan Uderhardt[2,3,4], and David B. Blumenthal[1,*]

[1]Biomedical Network Science Lab, Department of Artificial Intelligence in Biomedical Engineering, FAU, Erlangen, Germany
[2]Department of Medicine 3 – Rheumatology and Immunology, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Universitätsklinikum Erlangen, 91054 Erlangen, Germany
[3]Deutsches Zentrum für Immuntherapie, FAU, 91054 Erlangen, Germany
[4]Exploratory Research Unit, Optical Imaging Competence Center Erlangen, FAU, 91058 Erlangen, Germany
[5]Department of Biology, FAU, Erlangen, Germany
[6]Max-Planck-Zentrum für Physik und Medizin, Erlangen, Germany
[7]Max Planck Institute for the Science of Light, Erlangen, Germany
[*]Corresponding authors: `anna.moeller@fau.de`, `david.b.blumenthal@fau.de`

Abstract

A basic task that frequently arises when analyzing single-cell data is to assess if there is a global distribution shift between the data profiles of cells from two different conditions. Widely used approaches to address this task such as visual inspection of two-dimensional representations or clustering-based workflows lack a solid statistical underpinning and are notoriously unstable and prone to confirmation bias. To promote more rigorous analysis, we here present the scverse-compatible Python tool scXMatch. scXMatch is based on a non-parametric graph-based test to quantify distribution shifts in arbitrary data spaces for which a suitable distance measure is available. We evaluated scXMatch on single-cell gene expression, chromatin accessibility, and imaging-derived cell morphology data, showing that it can robustly detect distribution shifts for different types of single-cell data. scXMatch thus aims to set a new standard in the single-cell biology field, replacing easy-to-manipulate semi-manual distribution shift quantification workflows by principled statistical testing.

1 Introduction

The following question arises in many studies in the field of single-cell biology: Is there a global distribution shift between profiles of cells of the same type in a target condition in comparison to a control condition (e.g., diseased versus healthy, treated versus untreated, knockout versus wild-type)? Depending on the type of data — single-cell RNA sequencing (scRNA-seq), single-cell ATAC sequencing (scATAC-seq), etc. — typical workflows to answer this question include:

- Visually inspect the data upon projection into a two-dimensional (2D) space, e.g., using principal component analysis (PCA) followed by popular dimensionality reduction techniques such as UMAP [McInnes et al., 2018] or t-SNE [Van der Maaten and Hinton, 2008]. Conclude that there

is a global distribution shift if the 2D representations of the cells from the different conditions are visually separated.

- Carry out joint clustering of the single-cell data for the two conditions, followed by cluster-level over-representation analysis. Conclude that there is a global distribution shift if the cells from the different conditions are unevenly distributed over the clusters.

- Run statistical methods to identify individual differential features in the high-dimensional data, i.e., single variables such as differentially expressed genes (DEGs) or differentially accessible peaks that behave differently between the two conditions. Conclude that there is a global distribution shift if there are many differential features.

- Train a machine learning (ML) model that aims to separate the cells from the two conditions based on their data. Conclude that there is a global distribution shift if the ML model can accurately separate the two conditions.

Although widely used, all of these approaches have known shortcomings, which are particularly well studied for scRNA-seq data. For instance, 2D representations of scRNA-seq data have been shown to poorly preserve distances in the original high-dimensional data space, casting doubt on the reliability of visual inspection of 2D data representations [Chari and Pachter, 2023]. Clustering-based workflows are problematic because clustering scRNA-seq and other single-cell data is notoriously sensitive to the choice of the clustering algorithm [Yu et al., 2022, Liang et al., 2024]. Similar problems apply to workflows that rely on differential feature counts or ML-based separability as also here, different subroutines (i. e., different ML models or different methods to identify differential features) often produce substantially different results [Soneson and Robinson, 2018, Squair et al., 2021]. More generally, all of these approaches are indirect, at least to some degree *ad hoc*, and therefore easily lead to confirmation bias: Many hyper-parameters and sub-routines have to be selected along the way, and it is almost always close to impossible to make principled and non-arbitrary decisions regarding all of these steps. Given enough trial and error, it is hence very often possible to achieve the "desired" results.

To address these problems, we here present scXMatch — a Python package to directly test for global distribution shifts between sets $\mathscr{X}_1 = \{\mathbf{x}_i \mid y_i = 1\}$ and $\mathscr{X}_0 = \{\mathbf{x}_i \mid y_i = 0\}$ of h-dimensional profiles $\mathbf{x}_i \in \mathbb{R}^h$ of cells i from a target condition $y_i = 1$ and control condition $y_i = 0$. scXMatch implements a runtime- and memory-efficient version of the cross-matching test proposed by Rosenbaum [2005]. The cross-matching test is a non-parametric approach to test the null hypothesis that two sets of multi-variate data points $\mathscr{X}_0$ and $\mathscr{X}_1$ were sampled from the same distribution $F_0 = F_1$. It is ideally suited to assess global distribution shifts in single-cell data, as it relies on a test statistic whose distribution under the null hypothesis can be computed exactly even when the data-generating distribution $F_0 = F_1$ is unknown (details in Results and Methods below). The only required input besides $\mathscr{X}_0$ and $\mathscr{X}_1$ is a suitable distance measure $d : \mathbb{R}^h \times \mathbb{R}^h \to \mathbb{R}_{\geq 0}$ to quantify dissimilarity between the cell profiles.

Despite these desirable properties and although there is an R implementation of the cross-matching test [Heller et al., 2024], we are not aware of a single study which uses it to assess global distribution shifts in single-cell omics data such as scRNA-seq or scATAC-seq data. One likely reason for this is that the original version requires to compute a minimum-weight maximum-cardinality matching (MWMCM) in a complete weighted graph, where each data profile $\mathbf{x}_i$ is represented by a node. This requires quadratic memory with respect to the number of data profiles, which makes out-of-the-box usage of the cross-matching test infeasible for large-scale single-cell datasets. For scXMatch,

we therefore developed a substantially more efficient version of the cross-matching test, where the complete graph is replaced by a sparse k-nearest neighbor (k-NN) graph, which reduces the test's memory requirements from quadratic to linear. A systematic evaluation on scRNA-seq, scATAC-seq, and imaging-derived morphodynamics data shows that scXMatch scales to realistically sized single-cell datasets and that it can reliably distinguish signal from noise, using experimental perturbation of varying strength as ground truth. To further encourage adoption within the single-cell biology research community, we implemented scXMatch as an scverse-compatible Python package and published it in the scverse ecosystem [Virshup et al., 2023]. scXMatch is available on GitHub (`https://github.com/bionetslab/scxmatch`) and Bioconda (`https://anaconda.org/bioconda/scxmatch`).

2 Results

2.1 Assessing global distribution shifts with scXMatch

The general purpose of scXMatch is to examine whether two groups $\mathscr{X}_0$ and $\mathscr{X}_1$ of single-cell data profiles are sampled from the same distribution. Figure 1 provides a high-level overview of scXMatch; see Methods and the original publication of the cross-matching test [Rosenbaum, 2005] for further details. In a first step, scXMatch constructs a weighted undirected connectivity graph G, where the node set corresponds to the set of all data profiles $\mathscr{X} = \mathscr{X}_0 \cup \mathscr{X}_1$ and edges connecting two cells i and j are weighted by the dissimilarity $d(\mathbf{x}_i, \mathbf{x}_j)$ of their profiles. Depending on available computational resources and the size of $\mathscr{X}$, the user can decide to construct G as the complete graph or as a k-NN graph. Next, scXMatch computes a MWMCM M in G, meaning that as many cells as possible are assigned to exclusive pairs, such that the sum of distances between the features of paired cells is minimized. The pairs $ij \in M$ are either iso-matches or cross-matches, depending on whether the paired cells i and j are from the same condition ($y_i = y_j$) or from different conditions ($y_i \neq y_j$). The number of cross-matches a_1 is used as a test statistic, and a P-value and a z-score are computed based on the distribution of the corresponding random variable A_1. Under the null hypothesis that $\mathscr{X}_0$ and $\mathscr{X}_1$ are sampled from the same distribution, the distribution of A_1 is exactly known. The smaller the number of cross-matches a_1, the smaller the likelihood of observing it under the null hypothesis. That is, a small number of cross-matches (and, consequently, a negative z-score) is indicative of a global distribution shift between $\mathscr{X}_0$ and $\mathscr{X}_1$.

A major advantage of scXMatch is that it is very flexible regarding its input. Although designed with single-cell data in mind, it can be used for any kind of multi-variate data, as long as a suitable dissimilarity measure d is available. We provide out-of-the-box support for all distance functions implemented in SciPy; custom dissimilarity measures can be provided by the user as callables. For scRNA-seq data, we follow the recommendation of Ji et al. [2023] to use the squared Euclidean distance defined over the top 2000 most variable genes as a default for d. For scATAC-seq data, we used the gene scores and did likewise. For the morphodynamics data, we use standardized Euclidean distance.

2.2 scXMatch scales to large datasets

Besides the dissimilarity measure d, scXMatch requires the number of neighbors k as an additional hyperparameter when run in k-NN mode. In principle, it makes sense to select k as large as possible, since smaller k more strongly restrict the solution space for the MWMCM M (see Methods). However, setting k to larger values increases the runtime and memory requirements of scXMatch. To

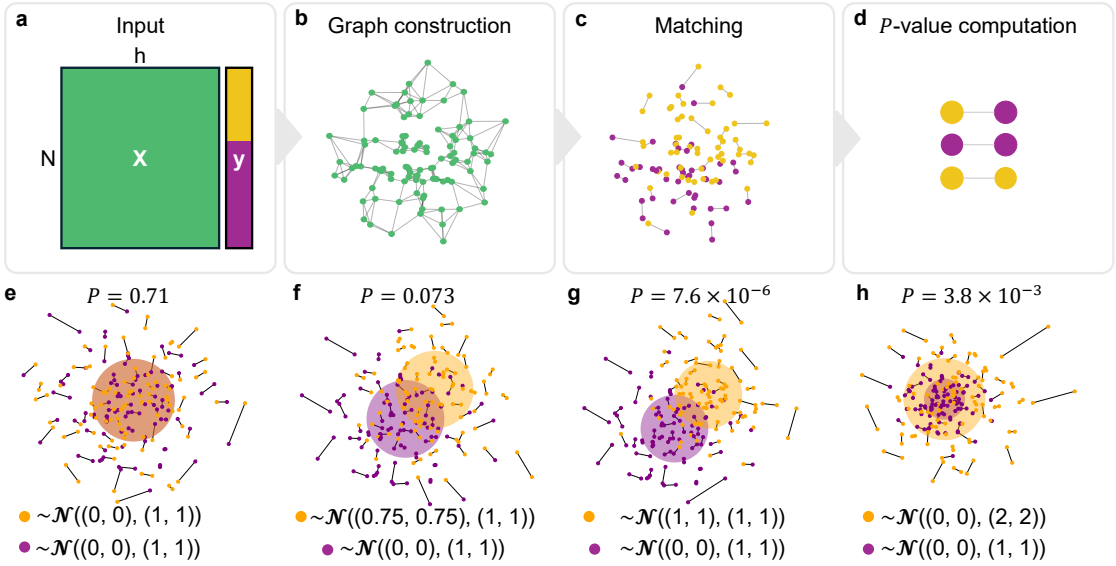

Figure 1. Overview of scXMatch. (a) As input, scXMatch requires: data profiles $\mathbf{x}_i \in \mathbb{R}^h$ for N cells $i = 1, \ldots, N$, binary condition labels $y_i \in \{0, 1\}$, here indicated by orange and purple, a measure d to quantify dissimilarity between the data profiles, a binary flag to specify if scXMatch should be run in k-NN mode, and a technical parameter k specifying the number of neighbors used for k-NN graph construction when run in k-NN mode. (b–c) scXMatch constructs an undirected graph G with N nodes (either a complete graph or a k-NN graph) and edge weights $d(\mathbf{x}_i, \mathbf{x}_j)$ for connected cells i and j and then computes a MWMCM M on G. (d) scXMatch counts the number a_1 of cross-matches $ij \in M$ that connect cells i and j with non-identical condition labels $y_i \neq y_j$ and computes a P-value and a z-score based on a_1. When a_1 is small, scXMatch rejects the null hypothesis that data profiles $\mathbf{x}_i$ and $\mathbf{x}_j$ with $y_i \neq y_j$ were sampled from the same distribution. (e–h) Exemplification of scXMatch (run in complete graph mode) using synthetic data sampled from bivariate normal distributions, with d set to the squared Euclidean distance. Significant global distribution shifts are detected both when there are differences in the means (f–g) and when there are only differences in the variances of the distributions, such that one population is drawn from a subspace of the other one (h). When data points from the two conditions are drawn from the same distribution, the P-value is non-significant (e).

provide a guideline on how to select k, we therefore simulated N data points $\mathbf{x}_i \in \mathbb{R}^h$ (corresponding to data profiles of individual cells in the context of single-cell data) for all combinations $(N, h) \in \{500, 1000, 2000, 5000, 10000, 20000, 50000\} \times \{10, 500, 1000, 2000\}$. On each of the resulting 28 simulated datasets, we ran scXMatch with varying $k \in \{10, 20, 50, 100, 200, 500, 1000, 2000, 5000\}$ and recorded the overall main memory usage, the overall runtime, and the relative coverage cov of the computed MWMCM M (as detailed in Equation (8), cov is the fraction of data profiles used towards the final P-value computation). Since the purpose of the runs on the simulated data was only to benchmark scXMatch's resource requirements, we simulated data with the respective dimensions by randomly drawing from a normal distribution. Each run was executed on a single node with 16 CPU cores and sufficient memory resources.

Figure 2 and Table S1 show the results. Overall, we identified main memory usage as the major limiting factor. To enable the user to select k based on the available memory, we fitted a linear regression model that estimates the total RAM usage based on the number of edges $|E|$ in the k-NN graph

G, using data from all scXMatch runs on the simulated data. Since the obtained model

$$\text{estimated total RAM [GB]} = \beta_0 + \beta_1 |E|, \text{ with } \beta_0 \approx 0.21 \text{ and } \beta_1 \approx 2.35 \cdot 10^{-7} \tag{1}$$

excellently fits the data (Figure 2a) and the number of edges $|E|$ is bounded from above by $|E| \leq Nk$ (how closely $|E|$ approaches this bound depends on the symmetry of the k-NN relationships), the user can select k based on the number of samples N and the available main memory as follows:

$$k = \left\lfloor \frac{\text{available RAM [GB]} - 0.21}{2.35 \cdot 10^{-7} N} \right\rfloor \tag{2}$$

Our Python implementation of scXMatch supports this automated selection of the hyperparameter k as a function of available RAM and dataset size.

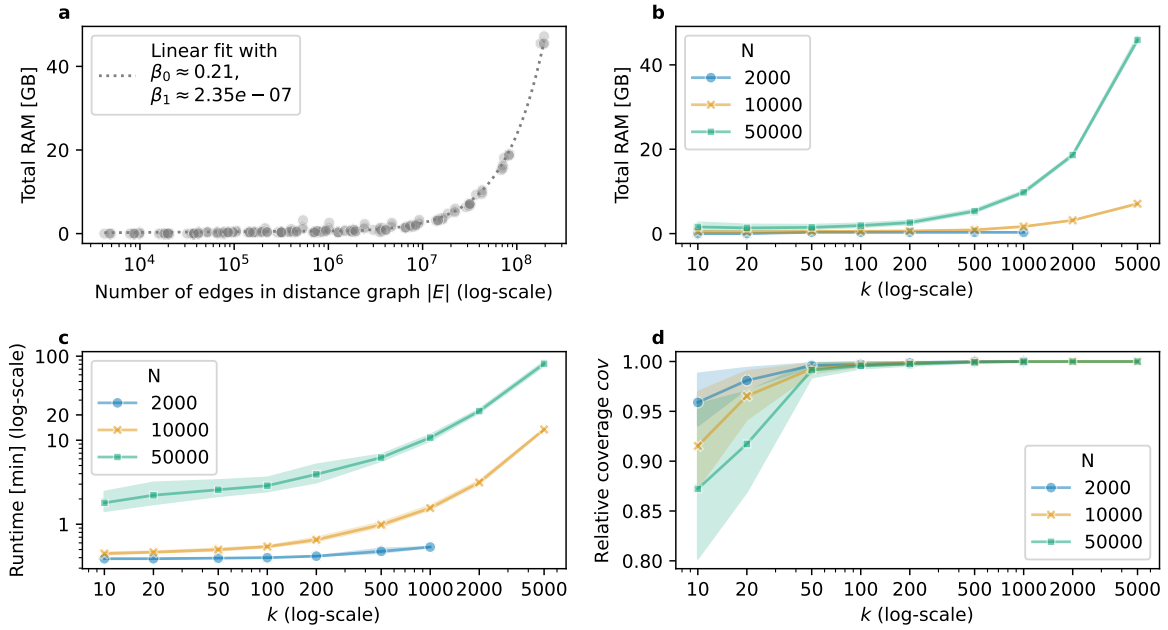

Figure 2. Results of the scalability tests. (a) Total memory usage as a function of the number of edges $|E|$ in the k-NN graph constructed by scXMatch. (b–d) Total memory usage (b), runtime (c), and relative coverage (d) as a function of k, for datasets with different numbers of samples N. Line plots and shaded areas in (b–d) depict means and 95% confidence intervals obtained for datasets with fixed N and k and varying dimensionality $h \in \{10, 500, 1000, 2000\}$.

Figure 2 also contains overviews on total memory usage (Figure 2b) and runtime (Figure 2c) as a function of k for different N. The test finished in under a minute for any value of k for datasets with $N = 2000$ samples, and in under 10 minutes with $k \leq 1000$ for datasets with $N = 50000$ samples. The dimensionality h of the input data only has a very small effect on scXMatch's resource requirements, as can be seen by the tight error margins of the line plots in Figure 2b and Figure 2c. The relative coverage, which indicates how many samples were covered by the matching, exceeds 99% for all N with $k \geq 100$ (Figure 2d). For all experiments on scRNA-seq data, we thus used $k = 100$, as this setup runs in reasonable time and memory and covers a convincing amount of samples. The scATAC-seq and morphodynamics datasets were smaller, which allowed us to run scXMatch in full graph mode.

2.3 scXMatch reliably detects global distribution shifts in scRNA-seq data

A method designed to quantify global distribution shifts in single-cell data should satisfy several key criteria to be considered robust and biologically meaningful. Firstly, it should be sensitive to biologically relevant differences, with output scores that scale with the underlying perturbation strength. Secondly, it should avoid false positives, i.e., it should return weak scores when comparing subsets drawn from the same distribution. Thirdly, its output should remain stable under subsampling from the compared distributions. To systematically assess these properties, we used ten public scRNA-seq datasets from studies where cells were exposed to perturbations of increasing strengths (Figure 3a): The datasets contain scRNA-seq data for cells of different cell lines at different time points after drug treatment [McFarland et al., 2020], for K562 cells upon single- and dual-guide CRISPR knockout [Norman et al., 2019], for embryonic fibroblasts at different time points after transcription factor-mediated cellular reprogramming [Schiebinger et al., 2019], and for different neuron subtypes in the prefrontal cortex of mice that were withdrawn for 48 hours or 15 days after cocaine addiction [Bhattacherjee et al., 2019]. We ran scXMatch with $k = 100$ and squared Euclidean distance as metric.

We defined three quantitative evaluation metrics: The first, a monotonicity index $\mathrm{MI}(s)$ (Figure 3b), measures whether a score s increases monotonically with increasing perturbation strength. The second metric, signal-stronger-than-noise ratio $\mathrm{SSNR}(s)$ (Figure 3c), compares scores derived from comparisons of condition against reference (signal) to those derived from comparisons within conditions (noise). The third metric, subsample variance $\mathrm{SV}(s)$ (Figure 3d), quantifies robustness by evaluating the variability of the score across different subsamples of the same condition. Using these metrics, we benchmarked scXMatch against a selection of commonly applied methods for determining global distribution shifts in scRNA-seq data. As very simple baseline, we included the number of DEGs between the two compared cell populations, with DEGs determined using edgeR [Robinson et al., 2010], DESeq2 [Love et al., 2014], and the Wilcoxon rank-sum test [Wilcoxon, 1992], respectively. Since edgeR and DESeq2 require bulk input, we employed a pseudo-bulk strategy by randomly grouping cells within each condition in batches of size 100. Moreover, we tested against the Augur score — a more advanced competitor approach which quantifies global distribution shifts as ML-based separability [Skinnider et al., 2021]. While Augur is typically used to assess cell-type-specific responsiveness to changing conditions, we here applied it in a global setting by evaluating its ability to discriminate between conditions in a single cell type.

The results of our benchmark are visualized in Figure 3e (see Tables S2–S7 for underlying numeric results). On five of the ten datasets, scXMatch is Pareto-optimal with regard to the three metrics, outperforming all competitors. That is, scXMatch performs at least as good or better than all competitors w. r. t. all three metrics. In terms of monotonicity, scXMatch achieved a perfect MI of 1.0 for nine out of ten datasets. Regarding SSNR, all tools achieved perfect scores. For SV, scXMatch produced the most stable scores five out of ten datasets (Bhattacherjee 2, McFarland 2, McFarland 3, Norman, and Schiebinger); on the others, the competitors where slightly more stable. Finally, we assessed the false positive rate (FPR) of scXMatch (for the competitor methods, we could not compute FPRs because they do not yield P-values): Out of all the 156 runs where we tested subsets of the same cell population against each other, scXMatch returned a significant raw P-value ($P < 0.05$) in 2 cases, leading to FPR $\approx 1.3\%$ for raw P-values and FPR $= 0\%$ after adjusting for multiple testing within each test group using Benjamini-Hochberg correction (Table S5).

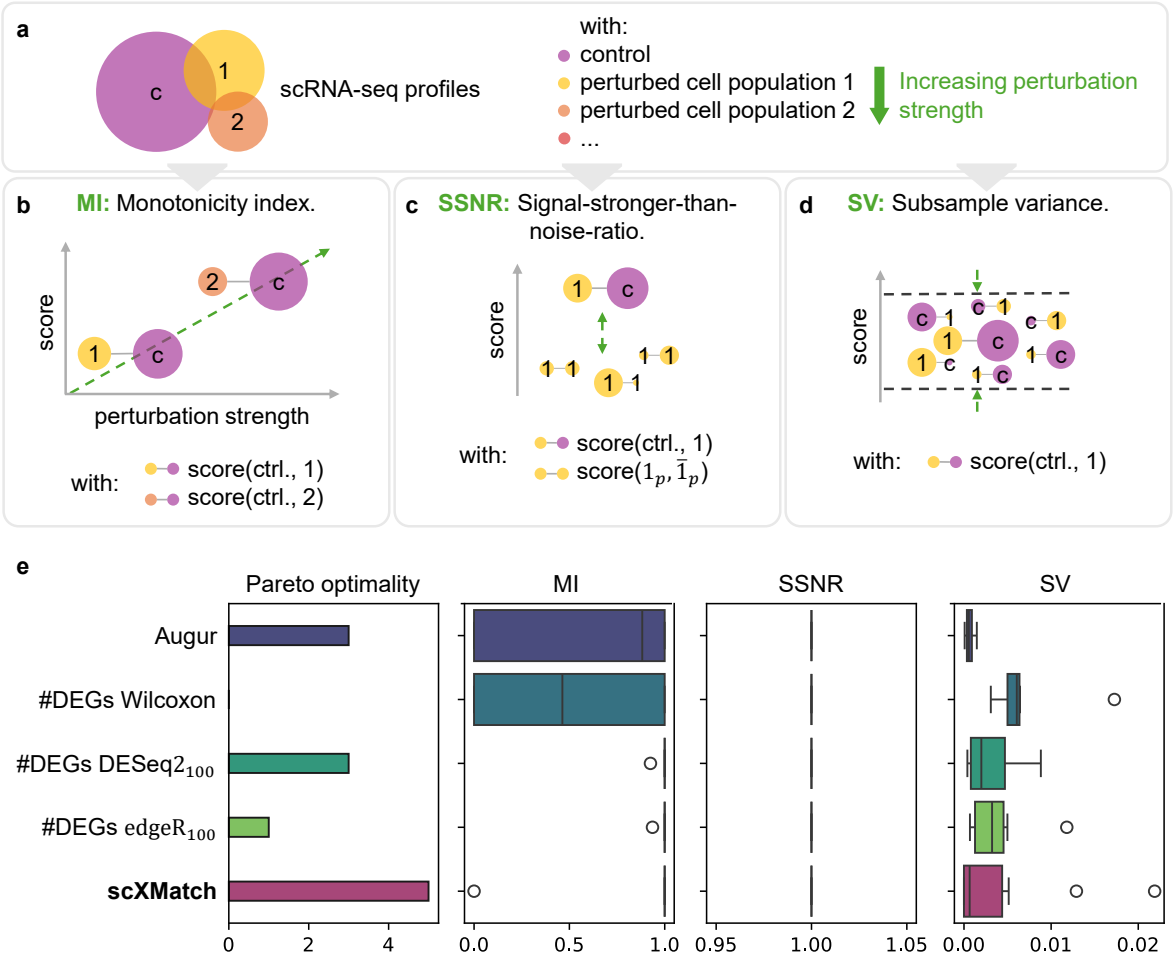

Figure 3. Benchmarking results for scXMatch. (a) We used ten public scRNA-seq datasets from cells that were subject to perturbations of increasing strengths and defined three metrics to compare our tool scXMatch against commonly used approaches that assess distribution shifts in scRNA-seq data. The green arrows indicate the desired tendency of the respective metric. (b) MI: Comparing cell populations with increasing perturbation strength against an unperturbed reference population should lead to increasing scores. (c) SSNR: Comparing a perturbed population against the reference should lead to a higher score than comparing disjunct subsets of the perturbed population against each other. (d) SV: Comparing different subsets of the reference against different subsets of the same perturbed condition should lead to similar scores, and thus a small variance of the results. (e) Benchmarking results of the five approaches on all ten scRNA-seq datasets. For five of them, scXMatch is Pareto-optimal with regard to the three metrics (left panel).

2.4 scXMatch implicitly removes outliers before assessing global distribution shifts

Running scXMatch in k-NN mode can lead to a matching coverage below 100% — particularly in datasets with a large number of samples. This observation raises the question if samples that are not included in the final matching share certain characteristics. Since scXMatch constructs the matching based on a k-NN graph, samples that do not reside in densely connected regions of the distance graph

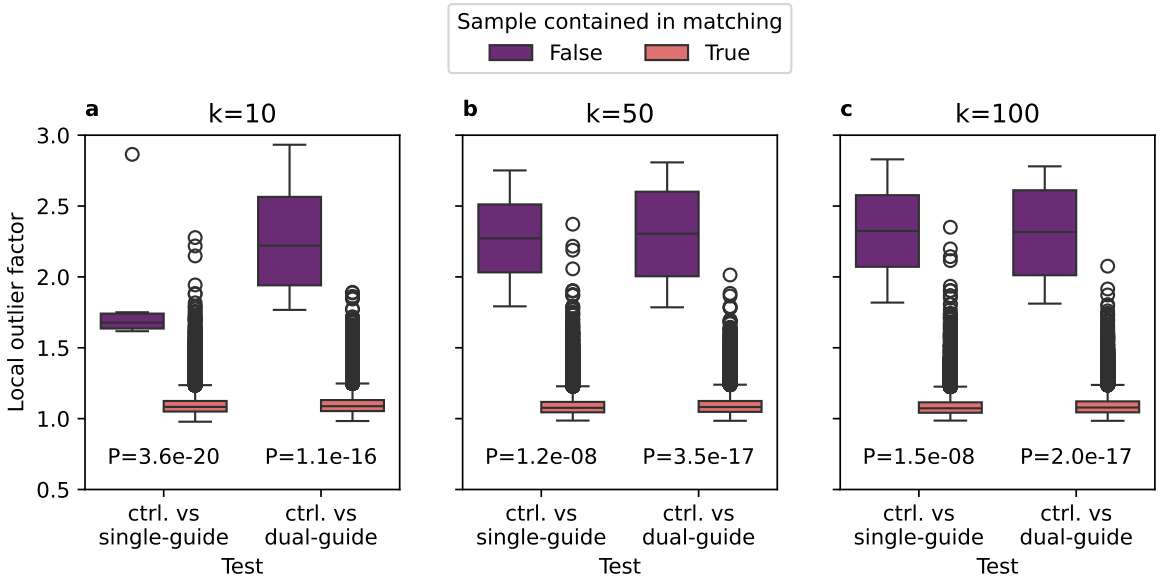

Figure 4. Local outlier factors, colored by inclusion in the MWMCMs obtained for the Norman dataset when using scXMatch with $k \in \{10, 50, 100\}$ to compare perturbed cell populations to untreated controls. The plots are annotated with *P*-values of the Wilcoxon rank-sum tests (alternative hypothesis: unmatched cells have larger LOFs than matched cells).

are more likely to be excluded from the matching. These samples may correspond to outliers.

To systematically investigate this hypothesis, we computed local outlier factors (LOFs) [Breunig et al., 2000] for each cell based on the full pairwise distance matrix, using the Norman dataset as test case, where scXMatch run in k-NN mode with $k \in \{10, 50, 100\}$ resulted in matching coverages below 100%. The LOF is a widely used density-based outlier score with LOF ≈ 1 for inliers and LOF > 1 for outliers. We then compared the distribution of LOFs between matched and unmatched cells, separately for each test group matched to its respective reference. Specifically, we performed one-sided Wilcoxon rank-sum tests with alternative hypothesis that the unmatched cells have larger LOFs than those included in the matching.

Figure 4 visualizes the LOF distributions, colored by whether or not a cell was included in the matching. Across all k and test groups, the hypothesis was supported with high statistical significance. These findings suggest that scXMatch, when run in k-NN mode, effectively performs an implicit outlier removal step by excluding cells that are not embedded within well-connected regions of the data manifold.

2.5 Changes in chromatin accessibility in scATAC-seq data identified by scXMatch reflect known T cell biology

scXMatch's utility extends to other single-cell omics modalities beyond scRNA-seq data. To demonstrate this, we analyzed scATAC-seq data from Mimitou et al. [2021], which maps open chromatin regions primed for gene activation. CRISPR–Cas9 perturbations of signaling components of the T cell repector (TCR) in human CD4+ T cells revealed context-dependent chromatin dynamics and accessibility. Due to the small sample sizes, we ran our tool in full graph mode. Using the squared Eu-

clidean distances for matching computation, scXMatch effectively ranked perturbation effects, with CD3E and ZAP70 knockouts yielding the most significant *P*-values (see Table 1). These upstream regulators are critical for structural TCR assembly (CD3E) and signal initiation/propagation (ZAP70), respectively, and drive chromatin remodeling by activating the transcription factors NFAT, AP-1, and NFκB, which prime enhancers of activation-associated genes. Their disruption caused a global loss of chromatin accessibility, reflecting their non-redundant roles in early TCR signaling [Shah et al., 2021]. In contrast, NFKB2 deficiency showed no significant effect, consistent with its role in sustaining late-phase responses rather than initiating chromatin remodeling [Sun, 2011]. Similarly, CD4 knockout had only a minimal effect on chromatin accessibility, since the anti-CD3/CD28 activation model used in this study bypasses the need for CD4's typical function in MHC II recognition [Damle and Doyle, 1989]. scXMatch hence consistently prioritizes biologically coherent hits (more significant *P*-values for upstream vs. downstream effectors), which underscores its value for identifying global chromatin shifts in perturbation screens.

Table 1. Benjamini-Hochberg corrected *P*-values obtained when running scXMatch on scATAC-seq data of human T cells subjected to targeted CRISPR-Cas9 knockout, sorted by decreasing significance.

Knockout, tested against control	Adjusted *P*-value
CD3E	1.2×10^{-25}
CD3E + CD4	1.1×10^{-20}
ZAP70	1.9×10^{-11}
NFKB2	0.44
CD4	1.0

2.6 scXMatch highlights importance of dynamic imaging for differentiating resident tissue macrophage populations

To exemplify the versatility of scXMatch beyond sequencing data, we used it to re-analyze data from our previous study on the morphology of differently stimulated resident tissue macrophages (RTMs) imaged *in vivo* in the peritoneum of mice [Schnitzerlein et al., 2025]. RTMs are characterized by their dynamic changes in cell shape, as they perform constant sampling of the environment [Uderhardt et al., 2019]. To describe those morphological characteristics, our previous study introduced a collection of 31 human-interpretable morphological RTM features, including 8 features that describe static RTM morphology and 23 features that quantify dynamic activity and adaption to stimuli (Figure 5a, Table S8). Overall, the resulting dataset contains data for around 350 individual RTMs in control mice and upon stimulation with 5 different stimuli: the anti-inflammatory stimulants macrophage colony stimulating factor (MCSF) and transforming growth factor β (TGF-β), the pro-inflammatory stimulants lipopolysaccharides (LPS) and interferon γ (IFN-γ), and the kinase inhibitor YM201636 (YM), which induces a halt of the sampling activity.

While our previous study identified distribution shifts in the 31-dimensional RTM morphology profiles between the different stimuli, we now sought to answer the question if the 8 static morphology quantifiers alone are sufficient to reliably distinguish RTM populations in different activation states. To do so, we used scXMatch to carry out pairwise comparisons between the different RTM populations, using all features and the static features only, respectively. The results are shown in Figure 5b. Overall, we observe that differences become less pronounced when relying on static features only, highlighting

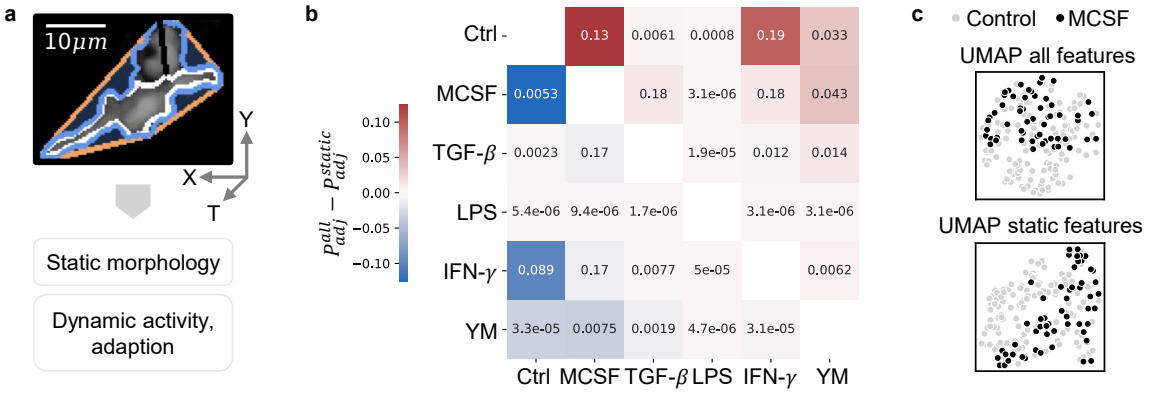

Figure 5. Importance of dynamic imaging data for distinguishing RTM populations. (a) Snapshot of an unperturbed, sampling macrophage *in vivo* taken from [Schnitzerlein et al., 2025]. Several morphological single-cell features (as schematically drawn on the image) were extracted. They characterize three categories of properties: static morphology, dynamic activity, or adaptation capabilities. While the static features can be calculated from a single image, the latter two require time-lapse images. (b) Benjamini-Hochberg-adjusted P-values returned by scXMatch for pairwise comparisons of RTM populations under different stimuli. Lower left triangle: adjusted P-values P^{all}_{adj} obtained when using all features. Upper right triangle: adjusted P-values P^{static}_{adj} obtained when using static features only. The heatmap is colored according to the difference of these two values (lower triangle minus upper triangle). Strong colors indicate large differences. Since the RTM dataset is small, we could run scXMatch in full graph mode. The standardized Euclidean distance was used as a distance measure. (c) UMAPs of the control versus MCSF-treated RTMs calculated on all and on only the static features, respectively.

the superiority of time-lapse *in vivo* imaging in capturing morphological RTM states over conventional imaging approaches that rely on fixated tissue samples [Zanier et al., 2015, Kim et al., 2024].

The strongest loss of differences can be observed for the comparison between MCSF-stimulated and control RTMs, where adjusted scXMatch P-values increase from 0.0053 when using all 31 morphological features (strongly significant) to 0.13 when using only the 8 static features (not significant). Figure 5c shows two-dimensional UMAP projections of the corresponding 31- and 8-dimensional data. While both projections suggest that the MCSF-stimulated RTMs span a subspace of the control RTMs, it is impossible to visually assess that one distribution shift is significant while the other one is not. This exemplifies the advantages of using scXMatch for assessing global distribution shifts in high-dimensional data instead of relying on visual inspection of two-dimensional projections.

3 Discussion

Statistical analysis of high-dimensional data is a common task in single-cell analysis. In this study, we introduced scXMatch — the first scalable, biologically validated, and easy-to-use implementation of a statistical test designed to quantify global distribution shifts in high-dimensional data. Instead of relying on dimensionality reduction methods such as PCA or UMAP, scXMatch bases its resulting statistics on a matching calculated in high-dimensional distance space. Unlike commonly used tools for differential feature analysis or clustering-based evaluation, scXMatch only requires a single tech-

nical hyperparameter, thereby reducing the risk of confirmation bias. In comparison to approaches like Augur scores, scXMatch assesses global distribution shifts rather than separability. This seemingly subtle difference implies that it can detect distribution shifts even if one cell population lie within a subspace of the other one (thus avoiding false negatives), and that it does not over-emphasize single features which may allow to separate the groups (thus avoiding false positives).

scXMatch implements a runtime- and memory-efficient version of Rosenbaum's cross matching test [Rosenbaum, 2005], making use of an efficient C++-based graph matching algorithms provided by the `graph-tool` package [Peixoto, 2014]. To further enhance memory and time efficiency, the user can specify — or let the tool automatically infer — a technical hyperparameter k, so that the matching is computed on a k-NN graph instead of the full distance graph. With the default $k = 100$, scXMatch runs in under a minute for 10000 samples, and in under 10 minutes for 50000 samples, with less than 5GB total RAM required. On datasets with a high number of samples, we showed that the matching construction via the k-NN graph implicitly removes outliers.

Aside from these technical aspects, scXMatch's results cohere with expected biological effects. On nine out of ten publicly available scRNA-seq datasets from perturbation experiments, the significance of the produced P-values was perfectly correlated with perturbation strength. We further introduced two metrics that describe specificity and robustness against subsampling, and scXMatch obtained Pareto-optimal results regarding these three criteria in five out of ten scRNA-seq datasets, outperforming all commonly used approaches. Beyond scRNA-seq data, we ran scXMatch tool on scATAC-seq data, where it returned results that perfectly align with known T cell biology, and on single-cell morphodynamic data of RTMs, where its results highlight the importance of time-lapse *in vivo* imaging to capture the functional state of RTMs.

It is important to stress that the scope of scXMatch is assessing the significance of distribution shifts between experimentally determined cell populations. Beyond this intended usage, scXMatch's P-values are less reliable. For instance, scXMatch should not be used to assess the significance of clusters that are *de novo* inferred from the single-cell data. This would amount to double dipping [Song et al., 2024], and scXMatch will thus almost always identify significant distribution shifts. Other dedicated methods exist to assess the significance of clustering results for scRNA-seq data [Grabski et al., 2023]. Caution also needs to be taken when comparing the results of two highly significant or two non-significant scXMatch tests. When two scXMatch P-values are both very small or when they are both non-significant, we observed that the order of the P-values does not always reflect the expected biological signal.

Our study opens up at least two interesting avenues for future work. Firstly, while scXMatch tests the null hypothesis that two cell populations $\mathscr{X}_0$ and $\mathscr{X}_1$ were sampled from the same distribution, a relevant adjacent question is if $\mathscr{X}_1$ was sampled from a subspace of $\mathscr{X}_0$. By incorporating outlier detection approaches to assess if all $\mathbf{x} \in \mathscr{X}_1$ are inliers w. r. t. $\mathscr{X}_0$, scXMatch could be extended to cover also this question. However, such an extension would be *ad hoc* and thus suffer from similar weaknesses as the existing indirect distribution shift quantification approaches which motivated us to develop scXMatch. Hence, an interesting challenge for future work is to design (or rediscover) a non-parametric statistical test which can be used for single-cell data and tests the null hypothesis that $\mathscr{X}_1$ was sampled from a subspace of $\mathscr{X}_0$. Secondly, the outlier removal side effect of scXMatch's k-NN graph approach could be exploited more systematically by allowing the user to specify the expected fraction r of outliers in the dataset and then setting the parameter k such that the relative coverage of the MWMCM computed on the k-NN graph is around $1 - r$. To implement such a strategy, it would be necessary to solve the (to the best of our knowledge) currently unsolved graph-theoretical question how to estimate the size of MWMCMs in k-NN graphs.

In sum, scXMatch is the first tool specifically designed to quantify global distribution shifts in single-cell data and thus fills a gap that was previously bridged with byproducts of methods originally designed for related but different purposes. Its value to the single-cell community lies in several aspects. Firstly, the tool can be used for almost any kind of single-cell data of all realistic sizes: The test itself is non-parametric, does not expect a specific distribution, and only requires the user to select a distance function to quantify cell dissimilarity. Various distance functions are supported out-of-the-box; custom distances can be provided by the user. Secondly, the concept of matching cells and then evaluating the resulting pairs is intuitively understandable even for users without statistics background. If desired, scXMatch returns the underlying matching, which enables inspection by the user and lends interpretability to scXMatch's P-values. Finally, scXMatch is interoperable with the scverse ecosystem and thus seamlessly interfaces with popular software for single-cell data analysis such as Scanpy [Wolf et al., 2018] and anndata [Virshup et al., 2024]. It is distributed via the Bioconda package manager [Grüning et al., 2018] and can be installed with just one line of code: `conda install scxmatch -c conda-forge -c bioconda`.

4 Methods

4.1 Rosenbaum's cross-matching test

Our tool scXMatch is based on the cross-matching test by Rosenbaum [2005]. To make this article self-contained, we here provide a high-level explanation of this test. Further details and proofs can be found in the original article.

The cross-matching test is designed for the following setup: We are given $N \geq 4$ samples $i \in \{1,\dots,N\}$ with binary condition labels $y_i \in \{0,1\}$. Of the N samples, $n \geq 2$ belong to a target condition $y_i = 1$ (i.e., $\sum_{i=1}^{N} y_i = n$) and the remaining $m = N - n \geq 2$ samples belong to a control condition $y_i = 0$. All samples are associated with multivariate data $\mathbf{x}_i$ from some data space $\mathbb{X}$ for which a suitable distance measure $d : \mathbb{X} \times \mathbb{X} \to \mathbb{R}_{\geq 0}$ is available. In the context of this article, we focus on the case where the samples are cells, $\mathbf{x}_i$ contains multivariate numeric data (e.g., gene expression or chromatin accessibility from scRNA-seq or scATAC-seq studies), and the condition labels y_i encode experimental conditions (e.g., wild type versus knockout, diseased versus healthy, treated versus untreated, etc.). However, in principle, the cross-matching test can be applied to any data as long as a distance measure d and binary condition labels are available.

Let $[N] = \{1,\dots N\}$, $\mathscr{X} = \{\mathbf{x}_i \mid i \in [N]\}$, and $\mathscr{X}_0 = \{\mathbf{x}_i \in \mathscr{X} \mid y_i = 0\}$, and $\mathscr{X}_1 = \{\mathbf{x}_i \in \mathscr{X} \mid y_i = 1\}$. Moreover, let F_0 and F_1 be two distributions over $\mathbb{X}$ such that $\mathscr{X}_0$ is sampled from F_0 and $\mathscr{X}_1$ is sampled from F_1. The null hypothesis of the cross-matching test is that $F_0 = F_1$. The test makes no assumptions on the distribution $F_0 = F_1$ under the null hypothesis, i.e., the distribution of the data under the null hypothesis may be unknown.

To compute its test statistic, the cross-matching test first constructs a weighted complete undirected graph $G = ([N], E, w)$ over the samples in the dataset, with the weight $w_{ij} \geq 0$ of an edge $ij \in E = \binom{[N]}{2}$ defined as the distance $w_{ij} = d(\mathbf{x}_i, \mathbf{x}_j)$ between the involved samples' data records. For even N (the case where N is odd is discussed in the next subsection), the cross-matching test then computes a minimum-weight perfect matching (MWPM) M for G, i.e., an edge set $M \subset E$ such that all nodes $i \in [N]$ are covered by exactly one edge $ij \in M$ (perfect matching) and M minimizes the overall edge weight $w(M) = \sum_{ij \in M} w_{ij}$ among all perfect matchings. For a fixed MWPM M and $l \in \{0,1,2\}$, let $A_l = |\{ij \in M \mid y_i + y_j = l\}|$ be the number of edges in M such that l incident nodes belong to condition 1. Note that $A_0 + A_1 + A_2 = |M|$ holds by construction and that we moreover have $N = 2|M|$,

$n = A_1 + 2A_2$, $A_2 = (n - A_1)/2$, and $A_0 = |M| - (n + A_1)/2$, since M is a perfect matching (i.e, A_1 determines A_0 and A_2). The cross-matching test uses A_1 as its test statistic — the number of crossing edges $ij \in M$ which connect samples i and j with different condition labels $y_i \neq y_j$. Under the null hypothesis $F_0 = F_1$, A_1 is distributed as follows:

$$Pr(A_1 = a_1) = \binom{N}{n}^{-1} \frac{|M|!}{a_0! a_1! a_2!} 2^{a_1} \tag{3}$$

To see why this is the case, we count the number of condition label assignments $(y_i)_{i=1}^N$ such that $\sum_{i=1}^N y_i = n$. There are $\binom{N}{n}$ such assignments, and, under the null hypothesis, they are independent of the matching M and thus all have an equal probability $\binom{N}{n}^{-1}$. Now we ask: How many condition label assignments are there such that $A_1 = a_1$ holds in addition to $\sum_{i=1}^N y_i = n$? To answer this question, consider one of the $|M|!$ different orderings

$$O = (\underbrace{i_1 j_1, \ldots, i_{a_0} j_{a_0}}_{O_0}, \underbrace{i_{a_0+1} j_{a_0+1}, \ldots, i_{a_0+a_1} j_{a_0+a_1}}_{O_1}, \underbrace{i_{a_0+a_1+1} j_{a_0+a_1+1}, \ldots, i_{|M|} j_{|M|}}_{O_2}) \tag{4}$$

of the edges $ij \in M$. Given O, we can construct a condition label assignment with $\sum_{i=1}^N y_i = n$ and $A_1 = a_1$ by setting $y_i = 0$ for all samples i covered by O_0, $y_i = 1$ for all samples i covered by O_2, and flipping a coin for each edge $ij \in O_1$ to decide if we set $y_i = 0$ and $y_j = 1$ or $y_i = 1$ and $y_j = 0$. There are 2^{a_1} different outcomes of the coin flipping subroutine, and, given a fixed outcome of this subroutine, two orderings which are identical up to permutation of O_0, O_1, and O_2 yield the same condition label assignment. This implies that, under the null hypothesis $F_0 = F_1$, A_1 is distributed according to Equation (3).

If $F_0 \neq F_1$, data elements $\mathbf{x}_i$ and $\mathbf{x}_j$ are sampled from different distributions for samples i and j with $y_i \neq y_j$, and so the weights w_{ij} are, on average, larger for crossing edges $ij \in E$ with $y_i \neq y_j$ than for pairs of samples with identical condition labels. Since the MWPM M prioritizes edges with small weights, the number A_1 of crossing edges in M is hence expected to be smaller under the alternative hypothesis $F_0 \neq F_1$ than under the null hypothesis. Consequently, we can reject the null hypothesis if the following P-value is small:

$$Pr(A_1 \leq a_1) = \sum_{a_1'=0}^{a_1} \binom{N}{n}^{-1} \frac{|M|!}{a_0'! a_1'! a_2'!} 2^{a_1'} \tag{5}$$

Rosenbaum [2005] further shows that, under the null hypothesis, A_1 has an expected value and a variance of

$$\mathbb{E}(A_1) = \frac{nm}{N-1} \qquad var(A_1) = \frac{2n(n-1)m(m-1)}{(N-3)(N-1)^2}, \tag{6}$$

which, for a given a_1, allows us to compute a z-score which is negative when the null hypothesis is rejected:

$$z(a_1) = (a_1 - \mathbb{E}(A_1))/\sqrt{var(A_1)} \tag{7}$$

4.2 Rosenbaum's handling of odd sample numbers

If N is odd, the graph $G = ([N], E, w)$ as introduced above does not have a perfect matching and so the probability distributions and P-values in Equations (3) and (5) are not well-defined. To handle this case, Rosenbaum [2005] suggests the following procedure:

1. Add dummy node ε to G and connect it to all real nodes $i \in [N]$ via edges of weight $w_{i\varepsilon} = 0$.

2. Compute a MWPM M_ε in the resulting augmented graph G_ε.

3. Discard the unique sample i such that $i\varepsilon \in M_\varepsilon$ and run the cross-matching test on graph G' containing the remaining $N-1$ samples (note that this also implies decreasing n by 1 if $y_i = 1$ holds for the discarded sample i).

We now provide an equivalent variant of this procedure, which will allow us to derive the speedup presented in the next subsection. Our variant is based on the notion of minimum-weight maximum-cardinality matchings (MWMCM). An MWMCM of a weighted undirected graph $G = ([N], E, w)$ is an edge set $M \subset E$ which covers each node $i \in [N]$ at most once, has maximum cardinality among all such edge sets (maximum-cardinality matching), and minimizes $w(M)$ among all maximum-cardinality matchings. MWMCMs exist for all weighted undirected graphs $G = ([N], E, w)$ and are perfect matchings whenever $G = ([N], E, w)$ has a perfect matching. We now make the following observation:

Observation 1. *Let N be odd and G, G', G_ε, and M_ε be constructed as detailed above and let M' be a MWPM for G'. Then M' is a MWMCM for G.*

Proof. By construction and since N is odd, any perfect matching for G' is a maximum-cardinality matching for G, and hence so is M'. Assume that there is a maximum-cardinality matching $M^\star$ for G with $w(M^\star) < w(M')$. Let $i^\star \in [N]$ be the unique sample left uncovered by $M^\star$ and let $M^\star_\varepsilon = M^\star \cup \{i^\star\varepsilon\}$. Then $M^\star$ is a perfect matching for $G^\star$ with $w(M^\star_\varepsilon) = w(M^\star) + w_{i^\star\varepsilon} = w(M^\star) < w(M') = w(M_\varepsilon)$, contradicting the choice of M_ε. □

Observation 1 leads to the following equivalent variant of Rosenbaum's procedure to handle odd N:

1. Compute a MWMCM M for G and collect the set $C(M) = \{i \in [N] \mid \exists e \in M : i \in e\}$ of samples covered by an edge $e \in M$.

2. Update the constants N, n, and m as follows: $N = |C(M)|$, $n = \sum_{i \in C(M)} y_i$, and $m = N - n$.

3. Compute the P-value and the z-score according to Equations (5) to (7), using the updated values of N, n, and m.

4.3 A more resource-efficient variant of the test based on k-nearest neighbor graphs

A major bottleneck of the original version of the cross-matching test is that it computes the matching M in the complete graph over the N samples. This implies that $O(N^2)$ calls to the distance function d are required to construct G and that we need $O(N^2)$ space to keep G in memory. While this lack of efficiency may be acceptable when N is small (e.g., because the samples correspond to patients as in the original paper's motivating example), it makes it impossible to use the cross-matching test out-of-the-box for the comparison of cell populations in single-cell datasets with thousands of cells.

To overcome this shortcoming, we observe that the alternative procedure to handle odd N explained at the end of the previous subsection can be run on arbitrary graphs G and not only when G is a complete weighted graph with an odd number of nodes. For our speedup of the cross-matching test, we hence simply construct G as a k-NN graph on $[N]$, where neighborhoods are defined in terms of the distance measure d. Although exact k-NN graph construction also requires $O(N^2)$ distance

computations, approximate k-NN graphs can be constructed efficiently, e.g., via the widely used NN-descent method [Dong et al., 2011]. Since, under the null hypothesis, connectivity in the k-NN graph is independent of the condition labels y_i, Equation (3) remains valid.

The main difference with respect to the original version is that there may now be several samples that are discarded before computing the P-value. Let

$$N_{\text{drop}} = N - |C(M)| \geq 0 \quad \text{and} \quad cov = \frac{|C(M)|}{2\lfloor N/2 \rfloor} \in [0,1] \tag{8}$$

denote, respectively, the number of discarded samples and the relative sample coverage of the matching M used towards the P-value computation. Smaller k lead to a larger N_{drop}, a smaller cov, and a more substantial gain in runtime and memory efficiency. As in the original version's procedure for handling odd N, under the null hypothesis, samples are discarded or retained independently of the condition labels y_i, and the discarded samples can be viewed as outliers in the multi-variate dataset $\mathscr{X}$ with respect to the distance measure d. We have $cov = 1$ whenever M is a perfect matching (for even N) or a matching covering all but one of the N samples (for odd N).

4.4 Implementation details

Since scXMatch is implemented as a Python package coherent to the guidelines of the scverse project [Virshup et al., 2023], the input data is expected to be provided in anndata format [Virshup et al., 2024]. Algorithmically, scXMatch is structured into three parts: first, the construction of the distance-weighted graph $G = ([N], E, w)$ with edge weights $w_{ij} = d(\mathbf{x}_i, \mathbf{x}_j)$, second, the computation of a MWMCM M in G, and third, the computation of the statistics given the resulting matching M and the group assignments $\mathbf{y}$. We rely on the following software packages for each of the steps:

Construction of distance-weighted graph. As described in the previous subsection, the package offers two different approaches for this purpose. The user can decide to either construct a complete graph, or a k-NN graph. The creation of the complete graph can be performed either on a CPU or a GPU. On a CPU, the package relies on `scipy.spatial.distance.cdist` [Virtanen et al., 2020]. If a GPU and a functioning `CuPy` [Okuta et al., 2017] installation are available, the computation is automatically run on the GPU. Calculating the distance matrix on a GPU leads to a significant speedup and is therefore strongly recommended for constructing the complete graph. The `cupyx.scipy.spatial.distance.cdist` implementation holds slightly fewer options for the chosen metric, but `sqeuclidean`, which scXMatch uses as a default for scRNA-seq data, is implemented. If the user decides to construct a k-NN graph, `scanpy.pp.neighbors` [Wolf et al., 2018] is used. To ensure that the dissimilarity is calculated on the actual omcis data and not on a lower-dimensional representation, we set the parameter `n_pcs=0`. `scanpy`'s k-NN graph construction relies on the NN-descent algorithm [Dong et al., 2011]. The resulting (sparse) distance-weighted adjacency matrix $(w_{ij})_{ij \in E}$ of G is efficiently stored as a `scipy.sparse.csr_matrix`.

Computation of MWMCM in distance-weighted graph. To compute an MWMCM M in G, we use `graph-tool` [Peixoto, 2014]. This Python package is especially suited for large-scale graph operations, as it is mostly implemented in C++. It is important to note that when constructing an undirected, weighted graph from a `scipy.sparse.csr_matrix` holding the weights, the `graph_tool.Graph` constructor ignores the lower diagonal of the matrix, assuming that the adjacency matrix is symmetric. As strictly speaking, the matrix constructed from the k-NN graph distances is not undirected — a

node x_i can be in a node x_j's k nearest neighbors while x_j not being in x_i's, which would lead to $w_{ij} = \infty$ but $w_{ji} < \infty$ in the weighted adjacency matrix of the k-NN graph — we modify the weights as

$$w'_{ij} = \min\{w_{ij}, w_{ji}\} \tag{9}$$

to construct an undirected graph that accurately preserves the neighboring relationships. The MWMCM M is retrieved with `graph_tool.topology.max_cardinality_matching`. This function internally calls the C++ Boost Graph Library implementation of Edmonds [1965]'s algorithm to compute maximum weight matchings, which runs in $O(N^3)$ time. To use this function, we hence have to transform our MWMCM instance into an equivalent instance of the maximum weight matching problem. We achieve this by further transforming the edge weights as follows:

$$w''_{ij} = \max_{i'j' \in E} w'_{i'j'} + 1 - w'_{ij} \tag{10}$$

Computation of P-values and z-scores. Given the matching M and the condition labels y_i, we implemented the remaining computations with basic Python and `numpy` functionalities. Since the computation of the final P-values leads to large numerators (see Equation (5)), we calculate the summands in logarithmic space and then aggregate the exponentials.

4.5 Datasets used for benchmarking

Simulated data. For the experiments to determine runtime and memory requirements, we simulate data and matching construction for all possible combinations of $N \in \{500, 1000, 2000, 5000, 10000, 20000, 50000\}$, $h = \{10, 500, 1000, 2000\}$, and $k \in \{10, 20, 50, 100, 200, 500, 1000, 2000, 5000\}$. The values for N and h were chosen such that they mimic common sizes of scRNA-seq datasets after pre-processing (e.g., filtering for highly variable genes). The chosen values for k aim at investigating how the matching coverage behaves for both small but efficient and bigger but less efficient k. Given the different parameter combinations, the data is randomly drawn from a normal distribution using NumPy's `random.normal` function. Note that the simluated data were only used to benchmark scX-Match's runtime and memory requirements, and are hence not required to realistically reflect patterns in scRNA-seq data.

Perturbation scRNA-seq data. We downloaded all datasets via the Python package `pertpy`. We `log1p`-transformed the data and selected the 2000 most variable genes using `scanpy`. The datasets have the following characteristics:

- The Bhattacherjee data contains scRNA-seq data from different mice [Bhattacherjee et al., 2019]. Three cohorts were on a steady level of cocaine for 15 days. Then, two cohorts experienced withdrawal for 48 hours and 15 days, respectively. From each of the groups, cells from the prefrontal cortext were collected and sequenced. We subset the dataset to the cell types with over 200 cells in each group, namely astrocytes, endothelial cells, and excitatory neurons. The number of cells per type and group is shown in Table 2. As we use the maintenance status as control, the effect strength is expected to be higher for the 15 day cohort.
- The McFarland dataset was generated using a novel MIX-Seq approach [McFarland et al., 2020]. Well-characterized cancer cell lines were treated with 13 different drugs (including targeted therapies, general cytotoxics, and highly selective compounds) and sequenced after

Table 2. Number of cells per type and group in the Bhattacherjee subsets.

Name	Cell type	Control	48h	15d
Bhattacherjee 1	Astrocytes	216	270	213
Bhattacherjee 2	Endothelial cells	478	477	753
Bhattacherjee 3	Excitatory neurons	1264	1688	4789

6 and 24 hours. We filter for (cell line, treatment) combinations with more than 200 cells for each time point and are left with 5 subsets, which include thyroid anaplastic carcinoma cells (CAL62), tongue squamos cell carcinoma cells (BICR31), and ovarian cancer cells (OAW42 and IGROV1), which were treated with either the strongly selective tool compound BRD3379, or the targeted cancer therapies Idasanutlin or Trametinib. We name them as shown in Table 3.

Table 3. Number of cells per type and group in the McFarland subsets.

Name	Cell line	Treatment	control	6h	24h
McFarland 1	BICR31	Idasanutlin	683	252	277
McFarland 2	BICR31	Trametinib	683	217	337
McFarland 3	CAL62	BRD3379	411	223	322
McFarland 4	IGROV1	BRD3379	457	206	274
McFarland 5	OAW42	BRD3379	340	201	296

- The Norman dataset originates from a pooled CRISPR-based screen designed to investigate the effects of single versus pairwise gene perturbations in K562 cells [Norman et al., 2019]. Upon grouping the dataset into control, single-guide, and dual-guide conditions, we obtained 11,835, 57,735, and 41,685 cells, respectively.

- The Schiebinger dataset was collected over an 18-day time course to study transcription factor–mediated cellular reprogramming [Schiebinger et al., 2019]. Between 1,956 and 11,088 mouse embryonic fibroblasts were profiled every 12 hours using single-cell RNA sequencing, resulting in a total of 163,532 cells across 35 time points.

Perturbation scATAC-seq data. We downloaded the Mimitou scATAC-seq data [Mimitou et al., 2021] from scPerturb [Peidli et al., 2024] and used the provided gene scores. The data consists of primary human CD4+ T cells subjected to targeted CRISPR–Cas9 perturbations of key genes involved in T cell receptor (TCR) signaling, namely CD3E (621 cells), CD4 (1167 cells), ZAP70 (1253 cells), and NFKB2 (1024 cells), as well as a double knockout of CD3E and CD4 (1167 cells) and a control set (1112 cells). Like for the scRNA-seq data, we `log1p`-transform the gene scores and subset the features to the 2000 most variable genes.

Macrophage morphodynamics. The *in vivo* morphodynamics data was retrieved from two-dimensional microscopy videos of resident tissue macrophages under different stimuli. To derive the features, the macrophages were segmented and tracked over time. Details on the imaging and feature extraction can be found in Table S8 and the original publication [Schnitzerlein et al., 2025].

4.6 Baseline methods

Number of differentially expressed genes. We identify DEGs between each test group and the reference group with `pertpy`'s implementation of three commonly used approaches:

- The Wilcoxon test is a non-parametric statistical test that compares the ranks of gene expression values between groups without assuming normal distribution [Wilcoxon, 1992]. It is commonly used for scRNA-seq data due to its robustness to outliers and ability to handle non-Gaussian distributions.
- EdgeR is a method based on negative binomial models that was originally developed for bulk RNA-seq [Robinson et al., 2010]. It accounts for biological variability and is well-suited for datasets with small sample sizes or overdispersed count data.
- PyDESeq2 is a Python implementation of the DESeq2 algorithm, which models gene counts using a negative binomial distribution and applies shrinkage estimators for dispersion and fold changes [Love et al., 2014, Muzellec et al., 2023]. It is designed to handle count-based expression data and provides robust statistical testing even in the presence of noise.

Augur scores. Augur is a cell type prioritization method that quantifies the separability of perturbed versus control cells within one cell type using supervised classification [Squair et al., 2021]. It trains a classifier (by default a random forest) to distinguish between conditions and evaluates performance via cross-validated AUC scores. High Augur scores indicate that transcriptional profiles under a perturbation are readily distinguishable from controls, making it a useful metric for comparing the magnitude of perturbation effects.

4.7 Evaluation metrics

Runtime and memory. We recorded runtime and memory requirements for all 28 simulated datasets with $(N,h) \in \{500, 1000, 2000, 5000, 10000, 20000, 50000\} \times \{10, 500, 1000, 2000\}$ and for each of the 9 different examined values $k \in \{10, 20, 50, 100, 200, 500, 1000, 2000, 5000\}$. All jobs were run on shared cluster nodes equipped with 2 × AMD EPYC 7502 CPUs (64 cores total) and 512 GB of RAM. Each job was allowed to use 16 CPU cores and up to 512 GB of RAM. We recorded the runtimes using Python's `time` module. The jobs' total RAM requirements where extracted from the log files returned by the Slurm workload manager.

Monotonicity w. r. t. perturbation strength. Let $(\mathscr{X}_0, \mathscr{X}_1, \ldots, \mathscr{X}_L)$ be one of the scRNA-seq perturbation datasets described above, where the set $\mathscr{X}_0$ contains the cells from the control condition, the sets $(\mathscr{X}_l)_{l=1}^L$ contain corresponding cells from the L perturbed conditions, and l increases with perturbation strength. Moreover, let s be one of the tested distribution shift scores (i. e., scXMatch P-value, Augur score, or number of differentially expressed genes) and $(s_{0,l})_{l=1}^L$ be the values obtained with s when comparing cells in $\mathscr{X}_0$ to cells in $\mathscr{X}_l$, for $l \in [L]$. We define a monotonicity index

$$\mathrm{MI}(s) = \binom{L}{2}^{-1} \sum_{l=1}^{L-1} \sum_{l'=l+1}^{L} \texttt{isStrongerOrEqual}(s_{0,l'}, s_{0,l}) \in [0,1], \tag{11}$$

where $\texttt{isStrongerOrEqual}(s_{0,l'}, s_{0,l}) \in \{0,1\}$ is set to 1 if and only if $s_{0,l'}$ is indicative of a stronger or equally strong distribution shift than $s_{0,l}$ (i. e., $\texttt{isStrongerOrEqual} = \texttt{isLessOrEqual}$ for the

P-values computed by scXMatch and `isStrongerOrEqual` = `isGreaterOrEqual` for the number of differentially expressed genes and the Augur scores). $\mathrm{MI}(s)$ counts the perturbation strength pairs where the differences in global distribution shifts w. r. t. the control condition quantified by s matches the biological expectation and divides this count by the total number of perturbation strength pairs. Values close to 1 indicate that the distribution shift score s is consistent with the biological expectation. We decided to assess monotonicity rather than strict monotonicity in order not to punish scores in situations where the effect of the biological perturbation is either very weak or very strong across all perturbation strengths l. To prevent that $\mathrm{MI}(\cdot)$ overemphasizes tiny differences, $\texttt{isStrongerOrEqual}(\cdot,\cdot)$ rounds the compared scores to two significant figures before comparing them (e. g., 0.0123 and 0.0121 are treated as equal).

False positive rate estimation for scXMatch and signal-stronger-than-noise ratio computation. Let $(\mathscr{X}_0, \mathscr{X}_1, \ldots, \mathscr{X}_L)$ be one of the scRNA-seq perturbation datasets. For $l = 0, \ldots, L$, we randomly sampled subsets $\mathscr{X}_l^p$ from $\mathscr{X}_l$ by assigning cells to $\mathscr{X}_l^p$ with probability $p \in P = \{0.1, 0.3, 0.5\}$. We then used all benchmarked distribution shift scores s to compare cells in $\mathscr{X}_l^p$ and $\mathscr{X}_l \setminus \mathscr{X}_l^p$, obtaining distribution shift scores $s_{l_p,\overline{l_p}}$. When s is the P-value computed by scXMatch, the scores $s_{l_p,\overline{l_p}}$ can be used to assess the false positive rate by counting the scores that reach statistical significance. We carried out this analysis using 0.05 as significance cutoff for both raw and adjusted P-values that were corrected for multiple testing within each test group via the Benjamini-Hochberg procedure, with the number of tests set to $(L+1) \cdot |P|$. Additionally, we assessed for all benchmarked distribution shift scores s if they are systematically stronger for comparisons between different conditions than for comparisons within the same condition. To this end, we computed the signal-stronger-than-noise ratio

$$\mathrm{SSNR}(s) = (L \cdot |P|)^{-1} \sum_{l=1}^{L} \sum_{p \in P} \texttt{isStrongerOrEqual}(s_{0,l}, s_{l_p,\overline{l_p}}) \in [0,1], \tag{12}$$

which assumes values close to 1 if the score s is consistently stronger or equal when comparing cells from a perturbed condition to cells from the control condition than when comparing cells from two subsets of the same perturbed condition.

Robustness. Finally, we assessed how robust the different distribution shift scores are when evaluated on different subsets of the control and perturbed conditions. For $l = 0, \ldots, L$, we randomly sampled subsets $\mathscr{X}_l^p$ from $\mathscr{X}_l$ by assigning cells to $\mathscr{X}_l^p$ with probability $p \in P' = \{0.1, 0.3, 0.5\}$. For all $(p, p') \in P' \times P'$, we then used the benchmarked distribution shift scores s to compare cells in $\mathscr{X}_0^p$ and $\mathscr{X}_l^{p'}$, obtaining distribution shift scores $s_{0_p,l_{p'}}$. Let $s_{\min} = \min\{s_{0_p,l_{p'}} \mid (l,p,p') \in [L] \times P' \times P'\}$ and $s_{\max} = \max\{s_{0_p,l_{p'}} \mid (l,p,p') \in [L] \times P' \times P'\}$. For a fixed perturbation condition $l \in [L]$, we computed the subsample variance

$$\mathrm{SV}(s,l) = \underset{(p,p') \in P' \times P'}{\mathrm{variance}} \left(\hat{s}_{0_p,l_{p'}} \right) \tag{13}$$

as the variance of re-scaled scores $\hat{s}_{0_p,l_{p'}}$ defined as follows:

$$\hat{s}_{0_p,l_{p'}} = \begin{cases} s_{0_p,l_{p'}} & \text{if } s \text{ is Augur score} \\ s_{0_p,l_{p'}}/2000 & \text{if } s \text{ is number of DEGs} \\ 1 - s_{0_p,l_{p'}} & \text{if } s \text{ is scXMatch } P\text{-value} \end{cases} \tag{14}$$

We applied this scaling to enforce that all re-scaled scores have the same range $[0,1]$, with scores close to 1 indicating a strong distribution shift. This ensures that $\mathrm{SV}(s,\cdot)$ and $\mathrm{SV}(s',\cdot)$ are comparable also for distribution shift scores s and s' with non-identical scales.

Pareto optimality. Equipped with the performance metrics $\mathrm{MI}(\cdot)$, $\mathrm{SSNR}(\cdot)$, and $\mathrm{SV}(\cdot)$, we determine Pareto optimality for all pairs of distribution shift scores s and scRNA-seq datasets $\mathscr{X}$ by assessing if s is Pareto-dominated by another distribution shift score s' on $\mathscr{X}$. The score s' Pareto-dominates s on $\mathscr{X}$ if and only if s' is better than s with respect to at least one of the three performance metrics (MI and SSNR: better = larger; SV: better = smaller) and not worse with respect to all of them. To avoid overemphasizing tiny differences, we round the performance metrics to two significant figures before comparing them.

Local outlier factors. To compute LOFs for the samples from the Norman dataset, we used scikit-learn's `LocalOutlierFactor`. We applied the same value of k and distance metric as used in the corresponding run of scXMatch. Since the module inherently returns negative LOFs, we inverted them to improve the interpretability of the results.

Data availability

- All used scRNA-seq is publicly available and can be downloaded via functions from pertpy's data module (`data.bhattacherjee()`, `data.mcfarland_2020()`, `data.norman_2019()`, and `data.schiebinger_2019_18day()`).
- The scATAC-seq data is publicly available and can be downloaded from scPerturb's Zenodo repository `https://doi.org/10.5281/zenodo.7058382`, file `MimitouSmibert2021.zip`.
- The macrophage morphodynamics data is publicly available and can be downloaded from Zenodo `https:doi.org/10.5281/zenodo.13929787`.

Code availability

- Source code of scXMatch: `https://github.com/bionetslab/scxmatch`.
- Easy-to-install packaged version: `https://anaconda.org/bioconda/scxmatch`.
- Data processing and evaluation scripts, evaluation results, and notebooks that were used to generate all plots in this manuscript: `https://github.com/bionetslab/scXMatch_paper`.
- Code that was used to retrieve morphodynamic features from the macrophage imaging data: `https://github.com/MiriamSchnitzerlein/MacrophageMorphodynamics`.

Competing interests

All authors declare no competing interests.

Author contributions statement

AM and DBB conceived and designed the study and drafted the manuscript. AM implemented scX-Match, carried out the analyses on the scRNA-seq and the scATAC-seq data, and visualized the results. MS and EG carried out the analyses on the morphodynamics data. SU, VZ, and DBB supervised this work. All authors provided feedback on results and revised and approved the manuscript.

Acknowledgments

DBB was funded by the German Research Foundation (DFG, 516188180), by the German Federal Ministry of Research, Technology and Space (BMFTR, 031L0309A), and by the Klaus Tschira Stiftung (00.003.2024). SU is supported by DFG (project-IDs 448121430, 505539112, 447268119), the Hightech Agenda Bavaria, and by an ERC starting grant (project-ID 101039438). VZ acknowledges the support by the DFG (German Research Foundation); project A7 within the Research Training Group "Immunomicrotope" (GRK 2740/447268119). The authors gratefully acknowledge the scientific support and HPC resources provided by the Erlangen National High Performance Computing Center (NHR@FAU) of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU). The hardware is funded by the German Research Foundation (DFG).

References

A. Bhattacherjee, M. N. Djekidel, R. Chen, W. Chen, L. M. Tuesta, and Y. Zhang. Cell type-specific transcriptional programs in mouse prefrontal cortex during adolescence and addiction. *Nat. Commun.*, 10(1):4169, 2019. doi: 10.1038/s41467-019-12054-3.

M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. LOF: Identifying density-based local outliers. *SIGMOD Rec.*, 29(2):93–104, 2000. doi: 10.1145/335191.335388.

T. Chari and L. Pachter. The specious art of single-cell genomics. *PLoS Comput. Biol.*, 19(8): e1011288, 2023. doi: 10.1371/journal.pcbi.1011288.

N. K. Damle and L. V. Doyle. Stimulation via the CD3 and CD28 molecules induces responsiveness to IL-4 in CD4+CD29+CD45R-memory T lymphocytes. *J. Immunol.*, 143(6):1761–1767, 1989. doi: 10.4049/jimmunol.143.6.1761.

W. Dong, C. Moses, and K. Li. Efficient k-nearest neighbor graph construction for generic similarity measures. In *WWW 2011*, pages 577–586, New York, NY, USA, 2011. ACM. doi: 10.1145/1963405.1963487.

J. Edmonds. Maximum matching and a polyhedron with 0, 1-vertices. *J. Nat. Bur. Stand. Sect. B*, 69: 125–130, 1965.

I. N. Grabski, K. Street, and R. A. Irizarry. Significance analysis for clustering with single-cell RNA-sequencing data. *Nat. Methods*, 20(8):1196–1202, 2023. doi: 10.1038/s41592-023-01933-9.

B. Grüning, R. Dale, A. Sjödin, B. A. Chapman, J. Rowe, C. H. Tomkins-Tinch, R. Valieris, J. Köster, and Bioconda Team. Bioconda: sustainable and comprehensive software distribution for the life sciences. *Nat. Methods*, 15(7):475–476, 2018. doi: 10.1038/s41592-018-0046-7.

R. Heller, D. Small, and P. Rosenbaum. *crossmatch: The Cross-Match Test*, 2024. URL `https://CRAN.R-project.org/package=crossmatch`. R package version 1.4-0.

Y. Ji, T. D. Green, S. Peidli, M. Bahrami, M. Liu, L. Zappia, K. Hrovatin, C. Sander, and F. J. Theis. Optimal distance metrics for single-cell RNA-seq populations. *bioRxiv*, 2023. doi: 10.1101/2023.12.26.572833.

J. Kim, P. Pavlidis, and A. V. Ciernia. Development of a high-throughput pipeline to characterize microglia morphological states at a single-cell resolution. *eNeuro*, 11(7):ENEURO.0014–24.2024, 2024. doi: 10.1523/ENEURO.0014-24.2024.

X. Liang, L. Cao, H. Chen, L. Wang, Y. Wang, L. Fu, X. Tan, E. Chen, Y. Ding, and J. Tang. A critical assessment of clustering algorithms to improve cell clustering and identification in single-cell transcriptome study. *Brief. Bioinform.*, 25(1):bbad497, 2024. doi: 10.1093/bib/bbad497.

M. I. Love, W. Huber, and S. Anders. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.*, 15(12):550, 2014. doi: 10.1186/s13059-014-0550-8.

J. M. McFarland, B. R. Paolella, A. Warren, K. Geiger-Schuller, T. Shibue, M. Rothberg, O. Kuksenko, W. N. Colgan, A. Jones, E. Chambers, D. Dionne, S. Bender, B. M. Wolpin, M. Ghandi, I. Tirosh, O. Rozenblatt-Rosen, J. A. Roth, T. R. Golub, A. Regev, A. J. Aguirre, F. Vazquez, and A. Tsherniak. Multiplexed single-cell transcriptional response profiling to define cancer vulnerabilities and therapeutic mechanism of action. *Nat. Commun.*, 11(1):4296, 2020. doi: 10.1038/s41467-020-17440-w.

L. McInnes, J. Healy, and J. Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv [stat.ML]*, 2018. doi: 10.48550/arXiv.1802.03426.

E. P. Mimitou, C. A. Lareau, K. Y. Chen, A. L. Zorzetto-Fernandes, Y. Hao, Y. Takeshima, W. Luo, T.-S. Huang, B. Z. Yeung, E. Papalexi, P. I. Thakore, T. Kibayashi, J. B. Wing, M. Hata, R. Satija, K. L. Nazor, S. Sakaguchi, L. S. Ludwig, V. G. Sankaran, A. Regev, and P. Smibert. Scalable, multimodal profiling of chromatin accessibility, gene expression and protein levels in single cells. *Nat. Biotechnol.*, 39(10):1246–1258, 2021. doi: 10.1038/s41587-021-00927-2.

B. Muzellec, M. Teleńczuk, V. Cabeli, and M. Andreux. PyDESeq2: a python package for bulk RNA-seq differential expression analysis. *Bioinformatics*, 39(9), 2023. doi: 10.1093/bioinformatics/btad547.

T. M. Norman, M. A. Horlbeck, J. M. Replogle, A. Y. Ge, A. Xu, M. Jost, L. A. Gilbert, and J. S. Weissman. Exploring genetic interaction manifolds constructed from rich single-cell phenotypes. *Science*, 365(6455):786–793, 2019. doi: 10.1126/science.aax4438.

R. Okuta, Y. Unno, D. Nishino, S. Hido, and C. Loomis. CuPy: A NumPy-compatible library for NVIDIA GPU calculations. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS)*, 2017. URL `http://learningsys.org/nips17/assets/papers/paper_16.pdf`.

S. Peidli, T. D. Green, C. Shen, T. Gross, J. Min, S. Garda, B. Yuan, L. J. Schumacher, J. P. Taylor-King, D. S. Marks, A. Luna, N. Blüthgen, and C. Sander. scPerturb: harmonized single-cell perturbation data. *Nat. Methods*, 21(3):531–540, 2024. doi: 10.1038/s41592-023-02144-y.

T. P. Peixoto. The graph-tool python library. *figshare*, 2014. doi: 10.6084/m9.figshare.1164194. URL `http://figshare.com/articles/graph_tool/1164194`.

M. D. Robinson, D. J. McCarthy, and G. K. Smyth. edgeR: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140, 2010. doi: 10.1093/bioinformatics/btp616.

P. R. Rosenbaum. An exact distribution-free test comparing two multivariate distributions based on adjacency. *J. R. Stat. Soc. Series B Stat. Methodol.*, 67(4):515–530, 2005. doi: 10.1111/j.1467-9868.2005.00513.x.

G. Schiebinger, J. Shu, M. Tabaka, B. Cleary, V. Subramanian, A. Solomon, J. Gould, S. Liu, S. Lin, P. Berube, L. Lee, J. Chen, J. Brumbaugh, P. Rigollet, K. Hochedlinger, R. Jaenisch, A. Regev, and E. S. Lander. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, 176(4):928–943.e22, 2019. doi: 10.1016/j.cell.2019.01.006.

M. Schnitzerlein, E. Greto, A. Wegner, A. Möller, O. Aust, O. Ben Brahim, D. B. Blumenthal, V. Zaburdaev, and S. Uderhardt. Cellular morphodynamics as quantifiers for functional states of resident tissue macrophages in vivo. *PLoS Comput. Biol.*, 21(5):e1011859, 2025. doi: 10.1371/journal.pcbi.1011859.

K. Shah, A. Al-Haidari, J. Sun, and J. U. Kazi. T cell receptor (TCR) signaling in health and disease. *Signal Transduct. Target. Ther.*, 6(1):412, 2021. doi: 10.1038/s41392-021-00823-w.

M. A. Skinnider, J. W. Squair, C. Kathe, M. A. Anderson, M. Gautier, K. J. E. Matson, M. Milano, T. H. Hutson, Q. Barraud, A. A. Phillips, L. J. Foster, G. La Manno, A. J. Levine, and G. Courtine. Cell type prioritization in single-cell data. *Nat. Biotechnol.*, 39(1):30–34, 2021. doi: 10.1038/s41587-020-0605-1.

C. Soneson and M. D. Robinson. Bias, robustness and scalability in single-cell differential expression analysis. *Nat. Methods*, 15(4):255–261, 2018. doi: 10.1038/nmeth.4612.

D. Song, S. Chen, C. Lee, K. Li, X. Ge, and J. J. Li. Synthetic control removes spurious discoveries from double dipping in single-cell and spatial transcriptomics data analyses. *bioRxiv*, 2024. doi: 10.1101/2023.07.21.550107.

J. W. Squair, M. Gautier, C. Kathe, M. A. Anderson, N. D. James, T. H. Hutson, R. Hudelle, T. Qaiser, K. J. E. Matson, Q. Barraud, A. J. Levine, G. La Manno, M. A. Skinnider, and G. Courtine. Confronting false discoveries in single-cell differential expression. *Nat. Commun.*, 12(1):5692, 2021. doi: 10.1038/s41467-021-25960-2.

S.-C. Sun. Non-canonical NF-κB signaling pathway. *Cell Res.*, 21(1):71–85, 2011. doi: 10.1038/cr.2010.177.

S. Uderhardt, A. J. Martins, J. S. Tsang, T. Lämmermann, and R. N. Germain. Resident macrophages cloak tissue microlesions to prevent neutrophil-driven inflammatory damage. *Cell*, 177(3):541–555.e17, 2019. doi: 10.1016/j.cell.2019.02.028.

L. Van der Maaten and G. Hinton. Visualizing data using t-SNE. *J. Mach. Learn. Res.*, 9(11), 2008.

I. Virshup, D. Bredikhin, L. Heumos, G. Palla, G. Sturm, A. Gayoso, I. Kats, M. Koutrouli, Scverse Community, B. Berger, D. Pe'er, A. Regev, S. A. Teichmann, F. Finotello, F. A. Wolf, N. Yosef, O. Stegle, and F. J. Theis. The scverse project provides a computational ecosystem for single-cell omics data analysis. *Nat. Biotechnol.*, 41(5):604–606, 2023. doi: 10.1038/s41587-023-01733-8.

I. Virshup, S. Rybakov, F. J. Theis, P. Angerer, and F. A. Wolf. anndata: Access and store annotated data matrices. *J. Open Source Softw.*, 9(101):4371, 2024. doi: 10.21105/joss.04371.

P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods*, 17(3):261–272, 2020. doi: 10.1038/s41592-019-0686-2.

F. Wilcoxon. Individual comparisons by ranking methods. In *Springer Series in Statistics*, pages 196–202. Springer New York, New York, NY, 1992. doi: 10.1007/978-1-4612-4380-9_16.

F. A. Wolf, P. Angerer, and F. J. Theis. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol.*, 19(1):15, 2018. doi: 10.1186/s13059-017-1382-0.

L. Yu, Y. Cao, J. Y. H. Yang, and P. Yang. Benchmarking clustering algorithms on estimating the number of cell types from single-cell RNA-sequencing data. *Genome Biol.*, 23(1):49, 2022. doi: 10.1186/s13059-022-02622-0.

E. R. Zanier, S. Fumagalli, C. Perego, F. Pischiutta, and M.-G. De Simoni. Shape descriptors of the "never resting" microglia in three different acute brain injury models in mice. *Intensive Care Med. Exp.*, 3(1):39, 2015. doi: 10.1186/s40635-015-0039-0.