Gromov-Wasserstein optimal transport to align single-cell multi-omics data

Pinar Demetci*[1,2], Rebecca Santorella*[3], Björn Sandstede[3], William Stafford Noble[4,5], and Ritambhara Singh[1,2]

[1]*Department of Computer Science, Brown University*
[2]*Center for Computational Molecular Biology, Brown University*
[3]*Division of Applied Mathematics, Brown University*
[4]*Department of Genome Sciences, University of Washington*
[5]*Paul G. Allen School of Computer Science and Engineering, University of Washington*
[*]*Equal Contribution*

Abstract

Data integration of single-cell measurements is critical for our understanding of cell development and disease, but the lack of correspondence between different types of single-cell measurements makes such efforts challenging. Several unsupervised algorithms are capable of aligning heterogeneous types of single-cell measurements in a shared space, enabling the creation of mappings between single cells in different data modalities. We present Single-Cell alignment using Optimal Transport (SCOT), an unsupervised learning algorithm that uses Gromov Wasserstein-based optimal transport to align single-cell multi-omics datasets. SCOT calculates a probabilistic coupling matrix that matches cells across two datasets. The optimization uses k-nearest neighbor graphs, thus preserving the local geometry of the data. We use the resulting coupling matrix to project one single-cell dataset onto another via a barycentric projection. We compare the alignment performance of SCOT with state-of-the-art algorithms on three simulated and two real datasets. Our results demonstrate that SCOT yields results that are comparable in quality to those of competing methods, but SCOT is significantly faster and requires tuning fewer hyperparameters. The code is available at `https://github.com/rsinghlab/SCOT`

1 Introduction

Single-cell measurements provide a fine-grained view of the heterogeneous landscape of cells in a sample, revealing distinct subpopulations and their developmental and regulation trajectories across time. The availability of measurements capturing various properties of the genome, such as gene expression, chromatin accessibility, DNA methylation, histone modifications, and chromatin 3D conformation, has increased the need for data integration methods capable of combining these disparate data types.

Despite the importance of this task, the heterogeneity among single cells presents unique challenges. For example, due to technical limitations, it is hard to obtain multiple types of measurements from the same individual cell. Furthermore, when we measure very different properties of a cell—such as its transcriptional and 3D chromatin profiles—we cannot a priori identify correspondences between features in the two domains. Accordingly, integrating two or more single-cell data modalities requires methods that do not rely on either common cells or common features across the data types. This property of the data prevents the application of some existing single-cell alignment methods because they require some correspondence information, either among the cells or the features [1–4].

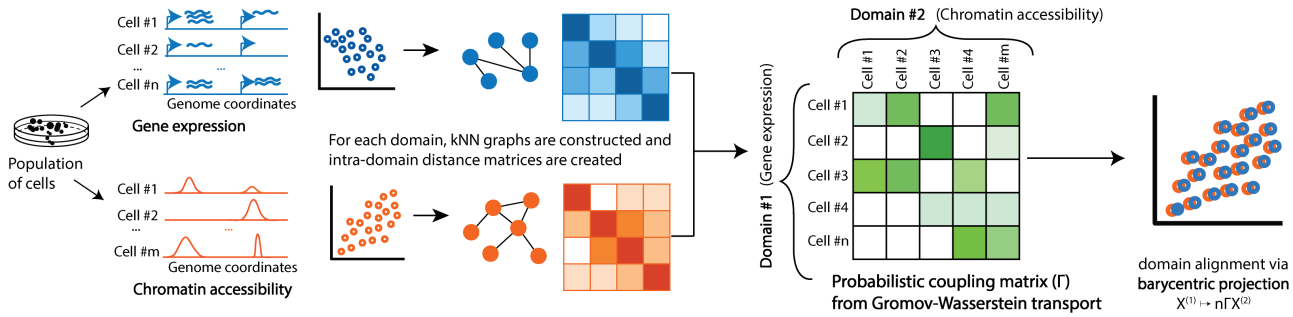


Figure 1: **Schematic of application of SCOT to single-cell multi-omics data alignment.** A population of cells is aliquoted for different single-cell sequencing assays in order to capture complementary aspects (e.g. gene expression and chromatin accessibility) of the molecular dynamics in single cells. Data obtained from these assays may exhibit different observed manifolds but share a common latent manifold. SCOT constructs k-NN graphs, where vertices represent cells, and Euclidean distances between them weigh the edges between the k-nearest neighbors. The SCOT algorithm finds a probabilistic coupling between the samples of each domain that will minimize the distance between the two intra-domain graph distance matrices. Barycentric projection uses this coupling matrix to project one domain onto another.

Some approaches have tried to align datasets in an entirely unsupervised fashion. One of the earliest attempts, the joint Laplacian manifold alignment (JLMA) algorithm, constructs eigenvector projections based on local k-nearest neighbor graph Laplacians of the data [5]. The generalized unsupervised manifold alignment (GUMA) [6] algorithm seeks a one-to-one correspondence between two datasets based on a local geometry matching term. Liu *et al.* [7] showed that these methods do not perform well on the single-cell alignment task. Specifically, the GUMA implementation was non-trivial to run, and JLMA gave poor a performance and did not scale well to larger values of k.

Liu *et al.* [7] proposed a manifold alignment algorithm based on the maximum mean discrepancy (MMD) measure, called MMD-MA, which can integrate different types of single-cell measurements. Another method, UnionCom [8], performs unsupervised topological alignment for single-cell multi-omics data. MMD-MA aims to match the global distributions of the datasets in a shared latent space, whereas UnionCom emphasizes learning both local and global alignments between the two distributions. Neither method requires any correspondence information either among samples or among the features of the different datasets. The papers demonstrate the state-of-the-art performance of the algorithms on simulated and real-world datasets. Although these results are encouraging, both MMD-MA and UnionCom require that the user specify four hyperparameters. In practice, selecting these hyperparameter values can be difficult and time-consuming in an unsupervised setting.

An emerging number of applications across different research areas [9, 10] are using optimal transport to learn a mapping between different data distributions. Optimal transport finds the most cost-effective way to move data points from one domain to another. One way to think about optimal transport is as the problem of moving a pile of sand to fill in a hole through the least amount of work. The optimal transport framework has been used in biological applications. Schiebinger *et al.* [11] use optimal transport to study how gene expression changes over time; they use regularized unbalanced optimal transport to compute differences in gene expression from one time point to the next. ImageAEOT [12] maps single-cell images to a common latent space through an autoencoder and then uses optimal transport to track cell trajectories. Its related work [13] uses autoencoders and optimal transport to learn transport maps between multiple domains. However, the application of this method to single-cell datasets requires some form of supervision, like class labels, to preserve the underlying structure during transport.

The classical optimal transport method requires datasets to be in the same metric space and is hard to implement for domains in different dimensions. Mémoli *et al.* [14] generalized optimal transport by using the Gromov-Wasserstein distance that compares metric spaces directly instead of comparing samples across spaces. In the natural language processing community, Alvarez *et al.* [10] used this approach to measure similarities between pairs of words across languages. They created uniform probability distributions on words in each language and used Gromov Wasserstein-based optimal transport to compute the distances between languages. As far as we are aware, the only biological application of Gromov-Wasserstein optimal transport comes from [15], that uses it to reconstruct the spatial organization of cells from transcriptional profiles. This approach assumes that the data consists of cells that were originally connected in tissue and that closer cells share similar transcriptional profiles but that the original spatial context and relationships among cells have been lost. With this setup, Nitzan *et al.*[15] use Gromov-Wasserstein optimal transport to map the cells to physical locations that preserve distances in the expression space.

In this paper, we present Single-Cell alignment using Optimal Transport (SCOT), an unsupervised learning algorithm that uses Gromov Wasserstein-based optimal transport to align single-cell multi-omics datasets (presented schematically in Figure 1). Like UnionCom, SCOT aims to preserve local geometry when aligning single-cell data. The algorithm achieves this by constructing a $k-$nearest neighbor graph for each dataset. SCOT then finds a probabilistic coupling between the samples of each dataset that minimizes the distance between the graph distance matrices produced by the k-NN graph. Finally, it uses the coupling matrix to project one single-cell dataset onto another through barycentric projection, thus aligning them. Unlike MMD-MA and UnionCom, our algorithm requires tuning of only two hyperparameters and is robust to the choice of one. We compare the alignment performance of SCOT with MMD-MA and UnionCom on three simulated and two real-world datasets. We demonstrate that SCOT aligns all the datasets as well as the state-of-the-art methods and converges $\sim$15 and $\sim$50 times faster than MMD-MA and UnionCom, respectively.

2 Method

SCOT uses Gromov Wasserstein-based optimal transport, which preserves local neighborhood geometry when moving data points. The output of this transport problem is a matrix of probabilities that represent how likely it is that data points from one space correspond to data points in the other space. These probabilities can then be used to project the data into the same space for alignment. In this section, we first introduce the formulation of optimal transport followed by its extension using the Gromov-Wasserstein distance. Finally, we present the details of our SCOT algorithm.

Without loss of generality, we present the case for two datasets. Let the two sets of points be $X = (x_1, x_2, \ldots, x_{n_x})$ from $\mathcal{X}$ and $Y = (y_1, y_2, \ldots, y_{n_y})$ from $\mathcal{Y}$. The datasets have n_x and n_y points, respectively. We do not require any correspondence information across the datasets but assume that there is some underlying shared structure so that the datasets can be aligned.

Optimal Transport The Kantorovich optimal transport problem seeks to find a minimal cost mapping between two probability distributions [16]. Referring back to the problem of moving a sand pile to fill in a hole, Kantorovich optimal transport allows us to split the mass of a grain of sand instead of moving the whole grain. For probability measures μ and ν defined on $\mathcal{X}$ and $\mathcal{Y}$, respectively, this optimal transport problem learns a minimal coupling π that attains

$$\min_{\pi \in \Pi(\nu,\mu)} \int_{\mathcal{X}\times\mathcal{Y}} c(x,y) d\pi(x,y), \tag{1}$$

where $c(x, y)$ is a cost function and $\Pi(\mu, \nu)$ is the set of couplings of μ and ν given by

$$\Pi(\mu, \nu) = \{\pi \in P(\mathcal{X} \times \mathcal{Y}) : \pi(A \times \mathcal{Y}) = \mu(A) \text{ for } A \subset \mathcal{X},\ \pi(\mathcal{X} \times B) = \nu(B) \text{ for } B \subset \mathcal{Y}\}. \quad (2)$$

Intuitively, the cost function says how many resources it will take to move x to y, and the coupling π assigns a probability that x should be moved to y for each x and y in the two spaces. Note that when the spaces of interest are both the same metric space with set $\mathcal{M}$, distance d, and cost function $c(x, y) = d(x, y)^p$, then the optimal transport distance (Equation 1) is equivalent to the $p-$th Wasserstein distance:

$$W^p(\mu, \nu) = \left(\inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{M} \times \mathcal{M}} d(x, y)^p d\pi(x, y) \right)^{\frac{1}{p}}. \quad (3)$$

Wasserstein distances measure the distances between probability distributions on a metric space and are commonly used in machine learning applications.

Since we want to align data, we work with discrete measures p and q over our data points, which we can write as

$$p = \sum_{i=1}^{n_x} p_i \delta_{x_i} \text{ and } q = \sum_{j=1}^{n_y} q_j \delta_{y_j},$$

where δ_{x_i} is the Dirac measure. Then, the cost function is given as a matrix $C \in \mathbb{R}^{n_x \times n_y}$, e.g. $C_{ij} = \|x_i - y_j\|$, and the set of possible couplings are the matrices

$$\Pi(p, q) = \{\Gamma \in \mathbb{R}_+^{n_x \times n_y} : \Gamma \mathbb{1}_{n_y} = p,\ \Gamma^T \mathbb{1}_{n_x} = q\}. \quad (4)$$

A discrete coupling Γ relates two measures p and q in a meaningful way: Each row Γ_i tells us how to split the mass of data point x_i onto the points y_j for $j = 1, \ldots, n_y$, and the condition $\Gamma \mathbb{1}_{n_y} = p$ requires that the sum of each row Γ_i is equal to the probability of sample x_i. The discrete optimal transport problem attempts to find a coupling that minimizes the cost of moving samples through the linear program:

$$\min_{\Gamma \in \Pi(p,q)} \langle \Gamma, C \rangle. \quad (5)$$

Although this problem can be solved with minimum cost flow solvers, it is usually regularized with entropy for more efficient optimization and empirically better results [17]. The addition of entropy diffuses the optimal coupling, meaning that more masses will be split. Thus, the optimal transport problem that is solved numerically is

$$\min_{\Gamma \in \Pi(p,q)} \langle \Gamma, C \rangle - \epsilon H(\Gamma), \quad (6)$$

where $\epsilon > 0$ and $H(\Gamma)$ is the entropy defined by

$$H(\Gamma) = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \Gamma_{ij} \log \Gamma_{ij}. \quad (7)$$

Equation 6 is a strictly convex optimization problem, and for some unknown vectors $u \in \mathbb{R}^{n_x}$ and $v \in \mathbb{R}^{n_y}$, the solution has the form $\Gamma^* = \text{diag}(u) K \text{diag}(v)$, with $K = \exp\left(-\frac{C}{\epsilon}\right)$, element-wise. This solution can be obtained efficiently via Sinkhorn's algorithm, which iteratively computes

$$u \leftarrow p \oslash Kv \text{ and } v \leftarrow q \oslash K^T u, \quad (8)$$

where $\oslash$ denotes element-wise division. This derivation immediately follows from solving the corresponding dual problem for Equation 6 [16].

Algorithm 1: Gromov-Wasserstein Alignment

Input: Datasets $X,\ Y$. Regularization ϵ. Number of neighbors k.
// Compute graph distances $D_x,\ D_y$;
p = Uniform(X), q = Uniform(Y);
$D_{xy} \leftarrow D_x^2 \mathbb{1}_{n_y}^T + \mathbb{1}_{n_x} q (D_x^2)^T$;
while *not converged* **do**
  // Compute cost matrix
  $\hat{D}_\Gamma \leftarrow D_{xy} - 2 D_x \Gamma D_y^T$;
  // Perform Sinkhorn iterations
  $u \leftarrow \mathbb{1}\,,\ K \leftarrow \exp\{-\hat{D}_\Gamma / \epsilon\}$;
  while *not converged* **do**
    $u \leftarrow p \oslash K v,\ v \leftarrow q^T \oslash K^T u$;
  end
  $\Gamma \leftarrow \text{diag}(u) K \text{diag}(v)$;
end
Return: Γ

Gromov-Wasserstein distance Classic optimal transport requires defining a cost function across domains, which can be difficult to implement when the domains are in different dimensions. Gromov-Wasserstein distance extends optimal transport by comparing distances between samples rather than directly comparing the samples themselves [10]. For this extension we need to assume we have metric measure spaces $(\mathcal{X}, d_x, \mu)$ and $(\mathcal{Y}, d_y, \nu)$, where d_x and d_y are distances on $\mathcal{X}$ and $\mathcal{Y}$, respectively [14]. Instead of defining a cost function between spaces as in classic optimal transport, Gromov-Wasserstein uses the difference between pairwise distances. Specifically, given a cost function $L : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, the Gromov-Wasserstein distance between μ and ν is defined by

$$GW(\mu, \nu) := \min_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} \int_{\mathcal{X}\times\mathcal{Y}} L(d_x(x_1, x_2), d_y(y_1, y_2)) d\pi(x_1, y_1) d\pi(x_2, y_2). \tag{9}$$

The main change from basic optimal transport (Equation 1) to Gromov-Wasserstein (Equation 9) is that we consider the effect of transporting pairs of points rather than single points. Intuitively, $L(d_x(x_1, x_2), d_y(y_1, y_2))$ now captures how transporting x_1 onto y_1 and x_2 onto y_2 would distort the original distances between x_1 and x_2 and between y_1 and y_2. This change ensures that the optimal transport plan π will preserve some local geometry. In the case of $L(x, y) = L_2(x, y) = \frac{1}{2}(x - y)^2$, Gromov-Wasserstein is a distance on the space of metric measure spaces [14].

For the discrete case, we can compute pairwise distance matrices D^x and D^y as well as the fourth order tensor $\mathbf{L} \in \mathbb{R}^{n_x \times n_x \times n_y \times n_y}$, where $\mathbf{L}_{ijkl} = L(D^x_{ik}, D^y_{jl})$. The discrete Gromov-Wasserstein problem is then defined by

$$GW(p, q) = \min_{\Gamma \in \Pi(p,q)} \sum_{i,j,k,l} \mathbf{L}_{ijkl} \Gamma_{ij} \Gamma_{kl}. \tag{10}$$

For each tuple (x_i, x_k, y_j, y_l), we are computing the cost of altering the pairwise distances between x_i and x_k when splitting their masses to y_j and y_l by weighting them by Γ_{ij} and Γ_{kl}, respectively. The summation can also be expressed as the inner product $\langle \mathbf{L}(D^x, D^y) \otimes \Gamma, \Gamma \rangle$. Equation 10 is now both non-linear and non-convex and involves operations on a fourth-order tensor, including the $\mathcal{O}(n_x^2 n_y^2)$ operation tensor product $L(D^x, D^y) \otimes \Gamma$ for a naive implementation. Peyré *et al.* show that for some choices of loss function this product can be computed in $\mathcal{O}(n_x^2 n_y + n_x n_y^2)$ cost [18]. In particular, for the case $L = L_2$,

the inner product can be computed by

$$\mathbf{L}(D^x, D^y) \otimes \Gamma = (D^x)^2 p \mathbb{1}_{n_y}^T + \mathbb{1}_{n_x} q^T ((D^y)^2)^T - D^x \Gamma (D^y)^T. \tag{11}$$

As in the classical optimal transport case, the coupling matrix can then be efficiently computed for an entropically regularized optimization problem:

$$GW(p, q) = \min_{\Gamma \in \Pi(p,q)} \langle \mathbf{L}(D^x, D^y) \otimes \Gamma, \Gamma \rangle - \epsilon H(\Gamma). \tag{12}$$

Larger values of ϵ lead to an easier optimization problem but also lead to a denser coupling matrix, meaning that more data points exhibit significant correspondences with one another. Smaller values of ϵ lead to sparser solutions, meaning that the coupling matrix is more likely to find the correct one-to-one correspondences for datasets where there are one-to-one correspondences. However, it also yields a harder (more non-convex) optimization problem [10].

Peyré *et al.* [18] propose using a projected gradient descent approach for optimization, where both the projection and the gradient are taken with respect to Kullback-Leibler divergence. These projections are computed via Sinkhorn iterations. Algorithm 1 presents the algorithm for $L = L_2$.

Single-Cell alignment using Optimal Transport (SCOT) Our method, SCOT, works as follows: First, we compute the pairwise distances on our data by using a geodesic distance as in [15]. To do this, we construct a k-nearest neighbor graph weighted by Euclidean distances within each dataset. Then we compute the shortest path distance on the graph between each pair of nodes. We set the distance of any unconnected nodes to be the maximum (finite) distance in the graph and rescale the resulting distance matrix by dividing by the maximum distance for numerical stability. Our approach is robust to the choice of k (Supplementary Section 1.4).

Since we do not know the true distribution of the original datasets, we follow [10] and set p and q to be the uniform distributions on the data points. With these graph distance matrices and marginal distributions, we solve for the optimal coupling Γ which minimizes Equation 12. To implement this method, we use the Python Optimal Transport toolbox (`https://pot.readthedocs.io/en/stable/`) [19]. The Sinkhorn iterations can often be unstable for small values of ϵ due to division by K, so we use the log stabilized version of the Sinkhorn iterations as proposed by [20, 21].

One of the major advantages of using Gromov-Wasserstein to align datasets is that we end up with a coupling matrix Γ with a probabilistic interpretation. In particular, the entries of the normalized row $n_x \Gamma_i$ are the probabilities that the fixed data point x_i corresponds to each y_j. However, to use the correspondence metrics previously used in the field to evaluate the alignment, we need to project the two datasets into the same space. The Procrustes approach proposed in in [10] does not generalize to datasets with different feature and sample dimensions, so we use a barycentric projection:

$$x_i \mapsto n_x \sum_{j=1}^{n_y} \Gamma_{ij} y_j. \tag{13}$$

This barycentric projection of point x_i is a weighted average of the y_j's, where the weight Γ_{ij} is the probability of correspondence between x_i and y_j. This projection averages over all the points. Thus, it has a tendency to center the projected data onto the mean of the dataset it is being projected on. Figure 1 presents the schematic of the SCOT algorithm.

3 Experimental Setup

Simulated datasets We follow Liu *et al.* [7] and benchmark our method on three different simulation schemes [1]. All three simulations contain two domains with 300 samples that have been projected non-linearly to 1000- and 2000-dimensional feature spaces, respectively. The first simulation is a bifurcated tree in two-dimensional space. The second simulation maps the branching structure onto a Swiss roll in three-dimensional space. The third simulation is a circular frustum in three-dimensional space (Supplementary Figure S1). Simulations are generated with known sample-wise correspondences, which are used to benchmark methods and evaluate their performance in recovering them. We Z-score normalize the features of all simulation datasets before running the alignment algorithms.

Single-cell multi-omics datasets We use two sets of single-cell multi-omics data to demonstrate the applicability of our model to real-world biological datasets. Both datasets are generated by co-assays; thus, we have known cell-level correspondence information for use in benchmarking.

The first dataset is generated using the sc-GEM assay [22], which simultaneously profiles gene expression and DNA methylation. The dataset (Sequence Read Archive accession SRP077853) is derived from human somatic cell samples undergoing conversion to induced pluripotent stem cells (iPSCs). This dataset was also used by Cao *et al.* [8] to demonstrate the performance of their UnionCom algorithm. The data dimensions are 177×34 for the gene expression data and 177×27 for the chromatin accessibility data.

The second dataset is generated by SNARE-seq [23], which links chromatin accessibility with gene expression. The data (Gene Expression Omnibus accession GSE126074) is derived from a mixture of human cell lines: BJ, H1, K562, and GM12878. We pre-processed the datasets following Chen *et al.* [23], as follows. We reduced data sparsity and noise in the ATAC-seq data by performing dimensionality reduction using the topic modeling framework cisTopic [24]. The dimensions of the RNA-seq data were reduced using PCA. The resulting input matrices for the SNARE-seq data were of size 1047×19 and 1047×10 for ATAC-seq and RNA-seq, respectively. Similar to the simulation datasets, we Z-score normalized all real-world datasets.

Baselines We compare SCOT with the two state-of-the-art unsupervised single-cell alignment methods MMD-MA [7] and UnionCom [8]. Note that none of these methods use any correspondence information for aligning the datasets.

Hyperparameter tuning To select hyperparameters, we ran each method over a grid of hyperparameters and selected the setting that yielded the maximal average FOSCTTM. For SCOT, the grid covers the regularization weight $\epsilon \in \{0.00001, 0.0001, 0.0002, 0.0003, ..., 0.1\}$ and number of neighbors $k \in \{5, 10, 20, 30, 40, \ldots n\}$, where n is the number of samples in the dataset. MMD-MA has four parameters to tune: the width $\sigma \in \{0.01, 0.1, 1.0, 10\}$ of the Gaussian for the initial kernel calculation, the weights $\lambda_1, \lambda_2 \in \{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}\}$ for the terms in the optimization problem, and the dimensionality $p \in \{3, 4, 5\}$ of the embedding space. UnionCom also requires the user to specify four hyperparameters: the number $k \in \{5, 10, 25, ..., n\}$ (with increments of 25 after $k = 25$) of neighbors in the graph, the dimensionality $p \in \{2, 5, 10\}$ of the embedding space, the trade-off parameter $\beta \in \{0.001, 0.005, 0.01, 0.5,\ 0.1, 0.5, 1, 5, 10\}$ for the embedding, and a regularization coefficient $\rho \in \{0.001, 0.005, 0.01, 0.5, 0.1, 0.5, 1, 5, 10\}$. While not related to the algorithmic formulation of

[1] `https://noble.gs.washington.edu/proj/mmd-ma/`

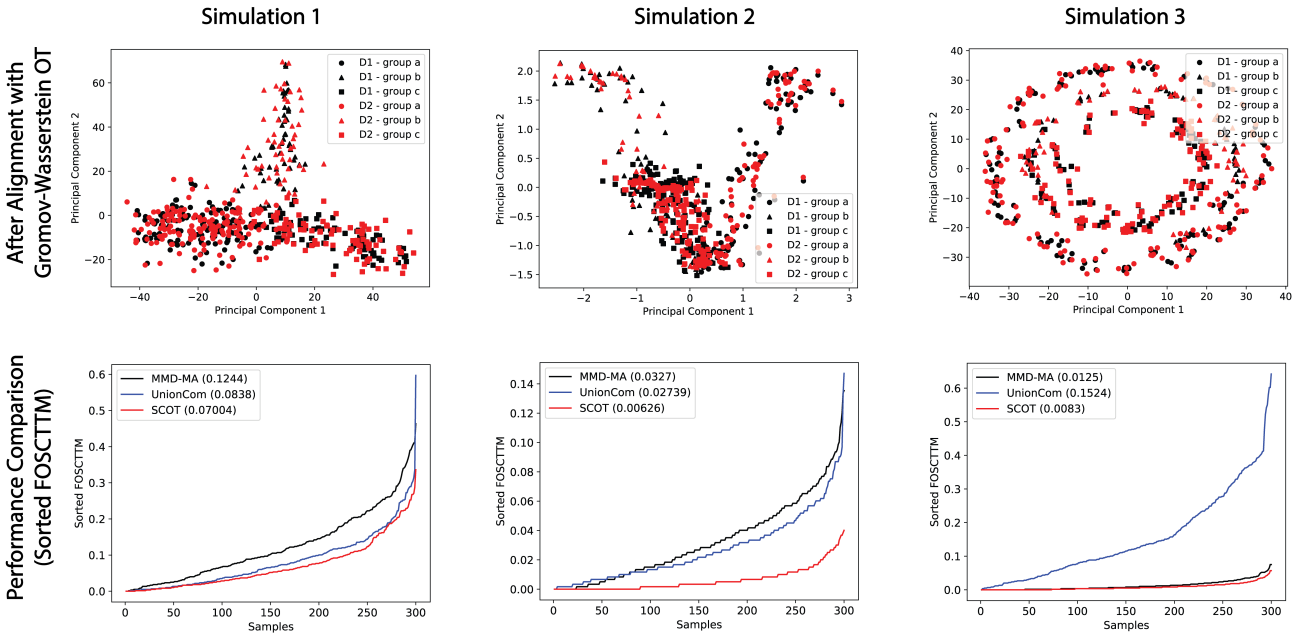


Figure 2: **Aligning simulated datasets.** Each column presents a different simulation. The top row presents our alignment (plotted in 2D using PCA), and the bottom row plots the "fraction of samples closer than the true match" (FOSCTTM) metric for MMD-MA, UnionCom, and SCOT. We also report the average FOSCTTM values in the legend. SCOT achieves the lowest average FOSCTTM values for all three simulations.

UnionCom, we also tuned the learning rate to achieve smoother convergence. We present alignment and runtime results for the best performing hyperparameters of SCOT, MMD-MA, and UnionCom.

Evaluation metrics All datasets have one-to-one sample-level correspondence information. We use this information solely to quantify the alignment performance of SCOT and the baselines. We use the evaluation metric previously introduced by Liu *et al.* [7] called "fraction of samples closer than the true match" (FOSCTTM). For each domain, we compute the Euclidean distances between a fixed sample point and all the data points in the other domain. Next, we compute the fraction of those distances that are closer to the sample than the distance to the true match. Next, we average these values for all the samples to give us an average FOSCTTM score. For perfect alignment, all samples would be closest to their true match, yielding a value of zero. Therefore, a lower average FOSCTTM corresponds to better alignment performance.

For the scGEM dataset, we also adopt a metric used by Cao *et al.* [8] called "label transfer accuracy." This metric assesses the alignment performance of the cell label assignment. Specifically, it measures the ability to correctly transfer sample labels from one domain to another based on their neighborhood in the aligned domain. As described in [8], we train a k-nearest neighbor classifier on one of the domains and predict the sample labels in the other domain. The label transfer accuracy is the percentage of correctly predicted labels, so it ranges from 0 to 100%, and higher values indicate better performance.

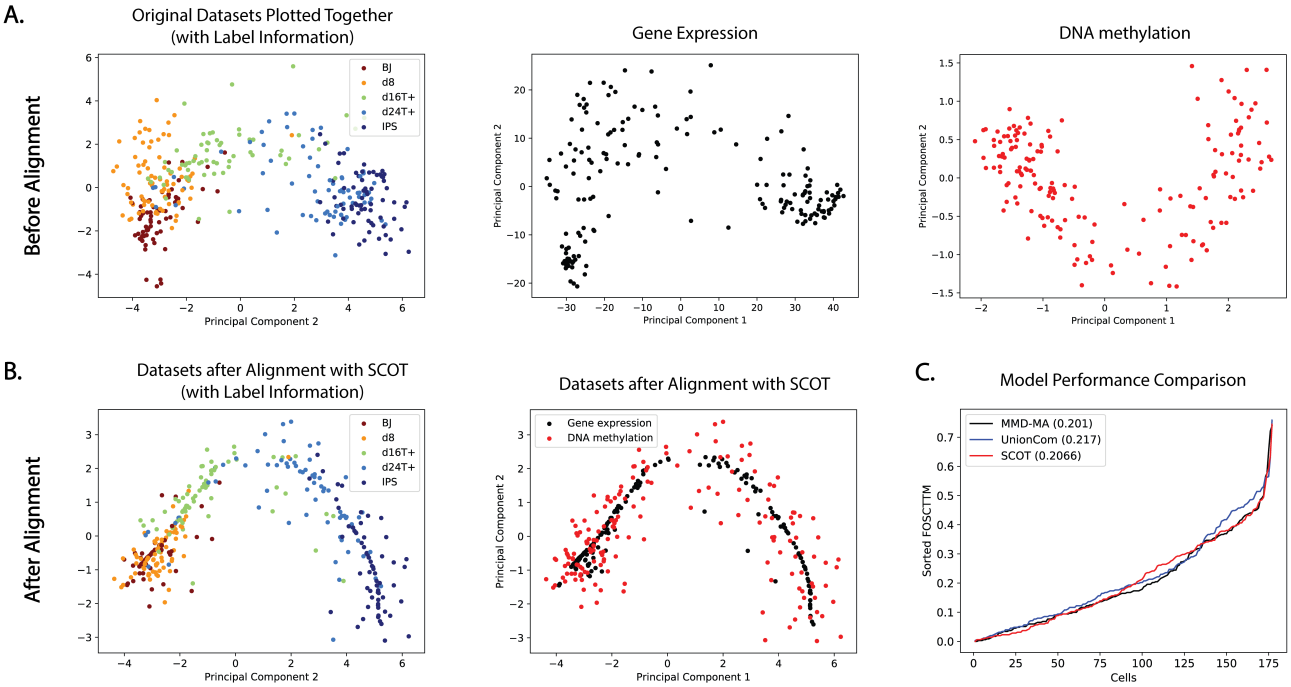


Figure 3: **Aligning the scGEM dataset. A.** The original scGEM datasets plotted in 2D using PCA. The first plot colors cells by label, and subsequent plots segregate cells by data type. **B.** SCOT alignment after performing optimal transport with barycentric projection. **C.** Comparison of the three algorithms, MMD-MA, UnionCom, and SCOT, based on FOSCTTM.

4 Results

SCOT successfully aligns the simulated datasets We first compare SCOT's performance with the baseline methods for the three simulation datasets. In Figure 2, we sort and plot the FOSCTTM score for each sample. We observe that SCOT achieves the lowest average FOSCTTM metric (averaged over all samples in the datasets) and demonstrates its ability to recover the correct correspondences in simulations.

SCOT gives state-of-the-art performance for single-cell multi-omics alignment Next, we apply our method to real-world single-cell sequencing assays and observe that SCOT gives comparable performance to the baseline methods. For scGEM data, the best FOSCTTM values are 0.201, 0.217, and 0.2066 for MMD-MA, UnionCom, and ScOT, respectively (Figure 3). Since the barycentric projection averages the data together, we observe that the expression data clusters near the mean of the manifold it is projected on (methylation data) in Figure 3(B).

As in [8], we use the label transfer accuracy metric to quantify how well the cells with the same label cluster together after alignment. For $k = 5$ (the default value used by Cao *et al.*), the label transfer accuracy values for MMD-MA, UnionCom, and SCOT are 0.5876, 0.5311, and 0.5650, respectively, when the chromatin accessibility dataset is used as the training set. For the training set comprised of gene expression, the values are 0.6384, 0.4689, and 0.6554, respectively. We report results for other values of k for $1 \leq k \leq 8$ in the Supplementary Figure S2.

Next, we compare all three methods for the SNARE-seq dataset (Figure 4). This dataset consists of a larger number of cells (1047) compared to scGEM (177). MMD-MA yields the best FOSCTTM,

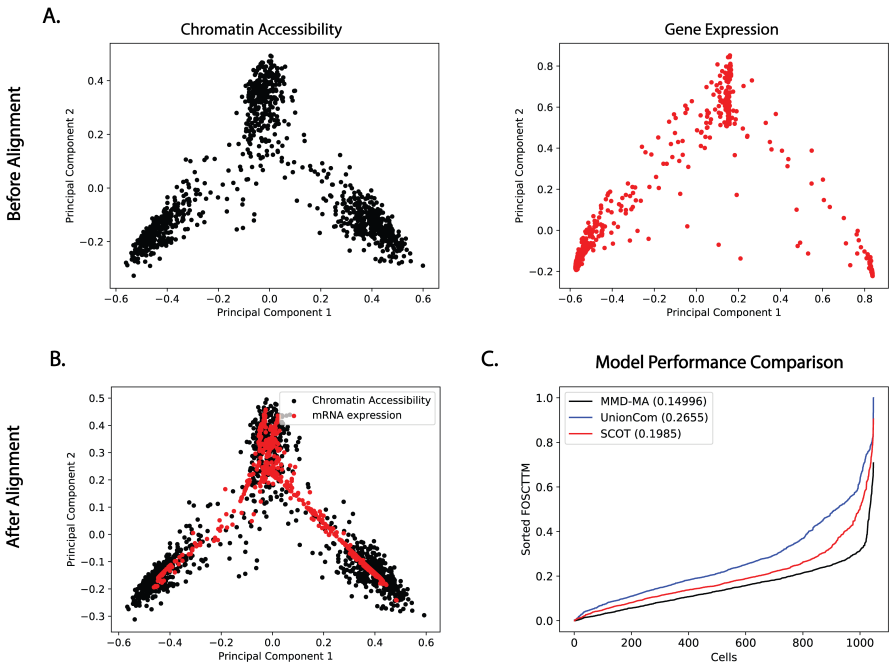


Figure 4: **Aligning SNARE-seq dataset. A.** The original SNARE-seq datasets plotted in 2D using PCA. **B**. SCOT alignment results after performing optimal transport with barycentric projection. **C.** Comparison of the three algorithms, MMD-MA, UnionCom, and, SCOT based on FOSCTTM.

	Sim. 1	**Sim. 2**	**Sim. 3**	**scGEM**	**SNAREseq**
SCOT	2.33	1.43	3.41	3.17	40.52
MMD-MA	30.06	29.69	28.84	16.12	547.71
UnionCom	525.85	442.19	302.69	143.60	2169.74
UnionCom (GPU)	117.72	112.41	109.73	70.21	345.37

Table 1: **Running times (in seconds) of SCOT, MMD-MA, and UnionCom for simulated and real datasets.** We observe that SCOT takes the least amount of time to converge.

followed by SCOT, with average FOSCTTM values of 0.1499 and 0.1985. UnionCom achieves lower performance with an average FOSCTTM value of 0.265.

A primary difference between MMD-MA and UnionCom versus SCOT is that, rather than projecting both the datasets to a lower-dimensional space, our method projects one dataset onto the other. To test whether the direction of the embedding matters, we ran SCOT in both directions for all datasets. In each case, we do not observe significant difference in performance between the two directions, with average FOSCTTM values of 0.0712 (Sim 1), 0.0063 (Sim 2), 0.0084 (Sim 3), 0.2220(scGEM), and 0.2281 (SNARE-seq).

SCOT is faster than other alignment algorithms We directly compared the running times of SCOT with the baseline methods for the best performing hyperparameters. We ran the CPU versions of the algorithms on an Intel i5-8259U CPU (base frequency 2.30GHz) with 16GB memory. UnionCom also has a GPU version that we ran on a single NVIDIA GTX 1080ti with VRAM of 11GB. We observe that SCOT converges $\sim$15, $\sim$50, and $\sim$10 times faster than MMD-MA, UnionCom, and UnionCom-GPU, respectively, for the largest SNARE-Seq dataset (Table 1).

5 Discussion

Integrating different single-cell modalities is an important task with challenges that require development of effective alignment algorithms. We have demonstrated that SCOT, which uses Gromov Wasserstein-based optimal transport to perform unsupervised integration of single-cell multi-omics data, performs well when compared to two state-of-the-art methods but in less time and with fewer hyperparameters.

To apply an evaluation metric and quantify the quality of alignment, we need to project the data into the same space. Here, we choose to use a barycentric projection to project one domain onto another, but there are various other ways to use the coupling matrix to infer alignment. For example, the coupling matrix could also be used with other dimension reduction methods such as t-SNE (as in UnionCom) to align the manifolds while embedding them both into new spaces. Additionally, depending on the application, a projection may not be required. For some downstream analyses, it may be sufficient to have probabilities relating the samples to one another. Our future work will focus on developing effective ways to utilize the coupling matrix and extend our framework to handle more than two alignments at a time.

We demonstrated the relative speed of convergence of SCOT. This speed benefit is further enhanced by the fact that, unlike MMD-MA and UnionCom which require tuning of four parameters, SCOT requires tuning of only two parameters. We also show (Supplementary Section 1.4) that SCOT is robust to the choice of k. In this way, SCOT dramatically reduces the hyperparameter search space, making application of the algorithm faster and easier.

Acknowledgments We are grateful to Yang Lu, Jean-Philippe Vert, and Marco Cuturi for helpful discussion of Gromov-Wasserstein optimal transport.

References

[1] Matthew Amodio and Smita Krishnaswamy. MAGAN: Aligning biological manifolds. 2018.

[2] Joshua D Welch, Velina Kozareva, Ashley Ferreira, Charles Vanderburg, Carly Martin, and Evan Z Macosko. Single-cell multi-omic integration compares and contrasts features of brain cell identity. *Cell*, 177(7):1873–1887, 2019.

[3] Joshua D Welch, Alexander J Hartemink, and Jan F Prins. Matcher: manifold alignment reveals correspondence between single cell transcriptome and epigenome dynamics. *Genome biology*, 18(1):138, 2017.

[4] Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M. Mauck III, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. Comprehensive integration of single-cell data. *Cell*, 77(7):1888–1902, 2019.

[5] Chang Wang and Sridhar Mahadevan. Manifold alignment without correspondence. In *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.

[6] Zhen Cui, Hong Chang, Shiguang Shan, and Xilin Chen. Generalized unsupervised manifold alignment. In *Advances in Neural Information Processing Systems*, pages 2429–2437, 2014.

[7] Jie Liu, Yuanhao Huang, Ritambhara Singh, Jean-Philippe Vert, and William Stafford Noble. Jointly embedding multiple single-cell omics measurements. *BioRxiv*, page 644310, 2019.

[8] Kai Cao, Xiangqi Bai, Yiguang Hong, and Lin Wan. Unsupervised topological alignment for single-cell multi-omics integration. *bioRxiv*, 2020.

[9] Alfred Galichon. A survey of some recent applications of optimal transport methods to econometrics. *Econometrics Journal*, 20(2), 2017.

[10] David Alvarez-Melis and Tommi S Jaakkola. Gromov-wasserstein alignment of word embedding spaces. *arXiv preprint arXiv:1809.00013*, 2018.

[11] Geoffrey Schiebinger, Jian Shu, Marcin Tabaka, Brian Cleary, Vidya Subramanian, Aryeh Solomon, Joshua Gould, Siyan Liu, Stacie Lin, Peter Berube, et al. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, 176(4):928–943, 2019.

[12] Karren D Yang, Karthik Damodaran, Saradha Venkatchalapathy, Ali C Soylemezoglu, GV Shivashankar, and Caroline Uhler. Autoencoder and optimal transport to infer single-cell trajectories of biological processes. *bioRxiv*, page 455469, 2018.

[13] Karren D Yang and Caroline Uhler. Multi-domain translation by learning uncoupled autoencoders. *arXiv preprint arXiv:1902.03515*, 2019.

[14] Facundo Mémoli. Gromov–wasserstein distances and the metric approach to object matching. *Foundations of computational mathematics*, 11(4):417–487, 2011.

[15] Mor Nitzan, Nikos Karaiskos, Nir Friedman, and Nikolaus Rajewsky. Gene expression cartography. *Nature*, 576(7785):132–137, 2019.

[16] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 2019.

[17] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems*, pages 2292–2300, 2013.

[18] Gabriel Peyré, Marco Cuturi, and Justin Solomon. Gromov-wasserstein averaging of kernel and distance matrices. In *International Conference on Machine Learning*, pages 2664–2672, 2016.

[19] Rémi Flamary and Nicolas Courty. Pot python optimal transport library, 2017.

[20] Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Scaling algorithms for unbalanced transport problems. *arXiv preprint arXiv:1607.05816*, 2016.

[21] Bernhard Schmitzer. Stabilized sparse scaling algorithms for entropy regularized transport problems. *SIAM Journal on Scientific Computing*, 41(3):A1443–A1481, 2019.

[22] Lih Feng Cheow, Elise T Courtois, Yuliana Tan, Ramya Viswanathan, Qiaorui Xing, Rui Zhen Tan, Daniel S Q Tan, Paul Robson, Loh Yuin-Han, Stephen R Quake, and William F Burkholder. Single-cell multimodal profiling reveals cellular epigenetic heterogeneity. *Nature Methods*, 13(10):833–836, 2016.

[23] Song Chen, Blue B Lake, and Kun Zhang. High-throughput sequencing of transcriptome and chromatin accessibility in the same cell. *Nature Biotechnology*, 37(12):1452–1457, 2019.

[24] Carmen Bravo González-Blas, Liesbeth Minnoye, Dafni Papasokrati, Sara Aibar, Gert Hulselmans, Valerie Christiaens, Kristofer Davie, Jasper Wouters, and Stein Aerts. cisTopic: cis-regulatory topic modelling on single-cell ATAC-seq data. 16(5):397–400, 2018.

1 Supplementary Information

1.1 Simulation Data Sets

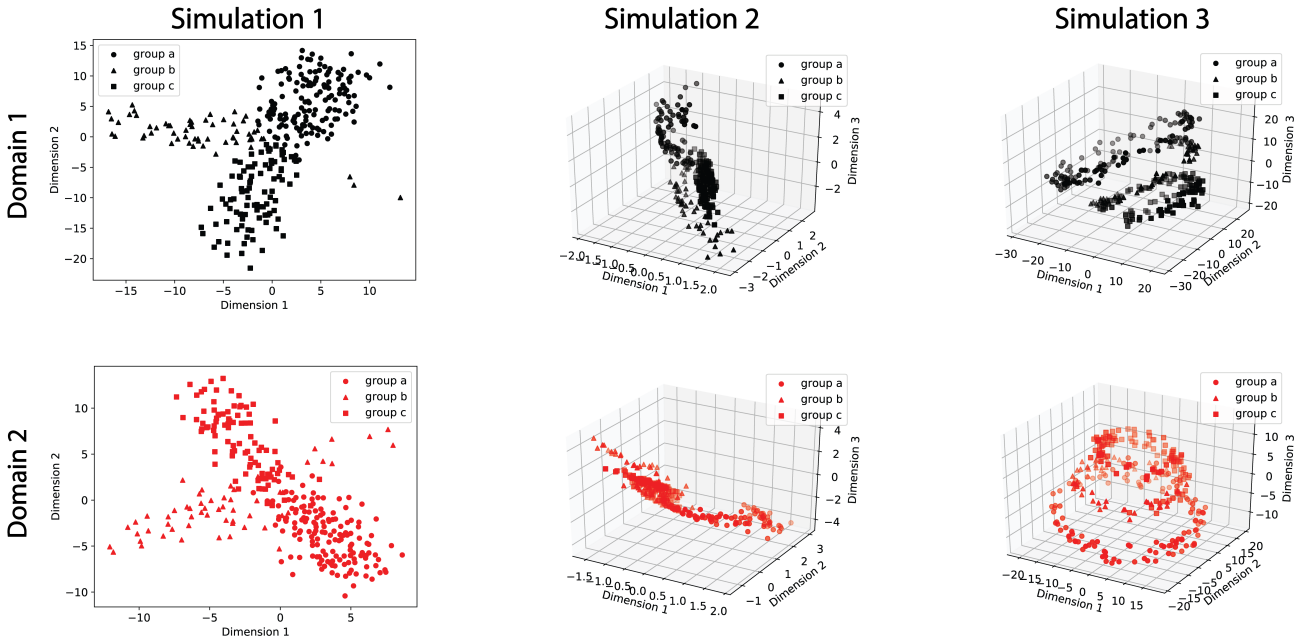


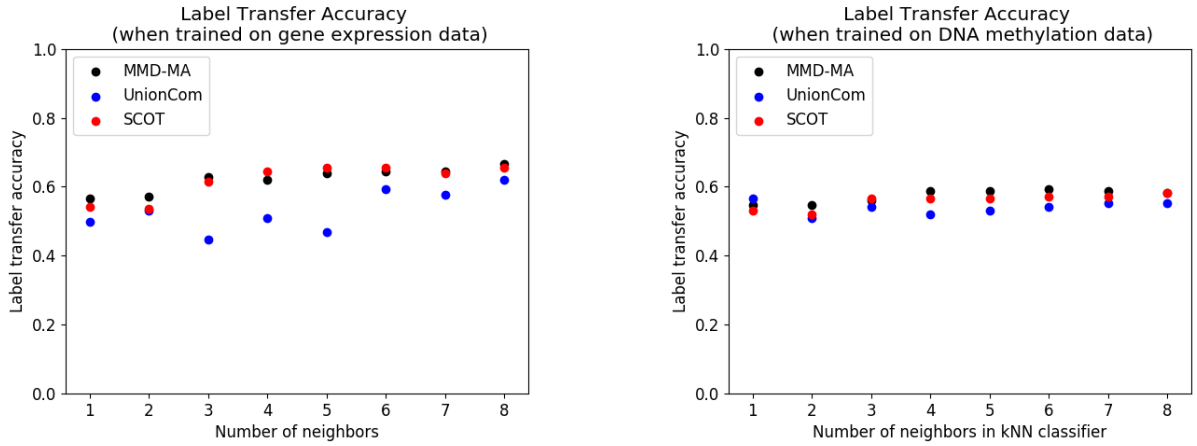
Figure S1: **Simulation data visualized before alignment.** Data was generated by Liu et al [7] and retrieved from `https://noble.gs.washington.edu/proj/mmd-ma/`. Each simulation set has two domains. Their MDS projections in two dimensional and three dimensional space are visualized here. The first set of simulations form a branched tree in two dimensional space (first column); the second set of simulations form Swiss roll in three dimensional space (second column); and lastly, the third set of simulations form a circular frustum.

1.2 Barycentric Projections in Both Directions

Dataset	Domain 1 onto Domain 2	Domain 2 onto Domain 1
Simulation 1	0.0700	0.0712
Simulation 2	0.0063	0.0059
Simulation 3	0.0083	0.0084
scGEM	0.2066	0.2220
SNARE-seq	0.2281	0.1985

Table 2: **Best mean FOSCTTM for each direction of the barycentric projection for all datasets.** The method is robust to the direction of the projection.

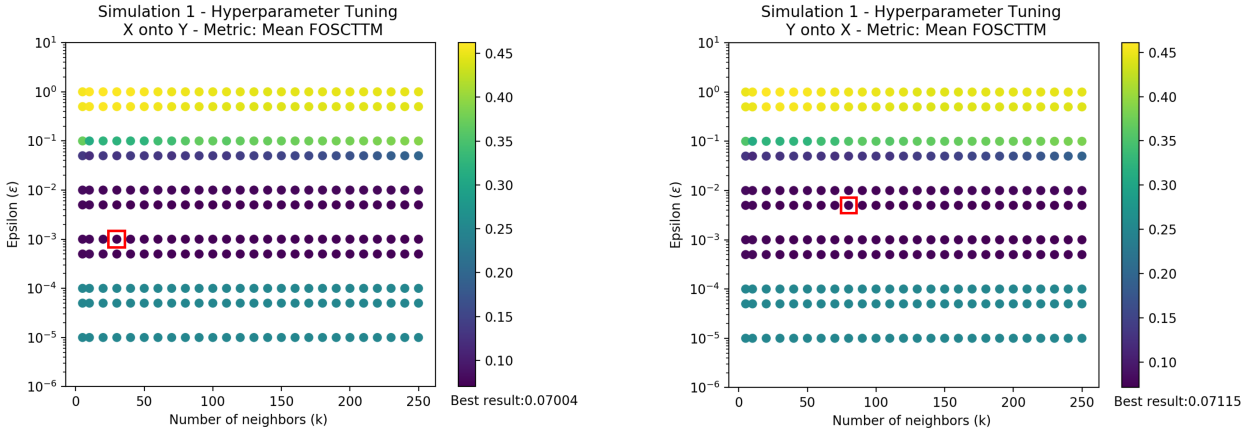
1.3 Label Transfer Accuracy for scGEM Data Set



(a) (b)

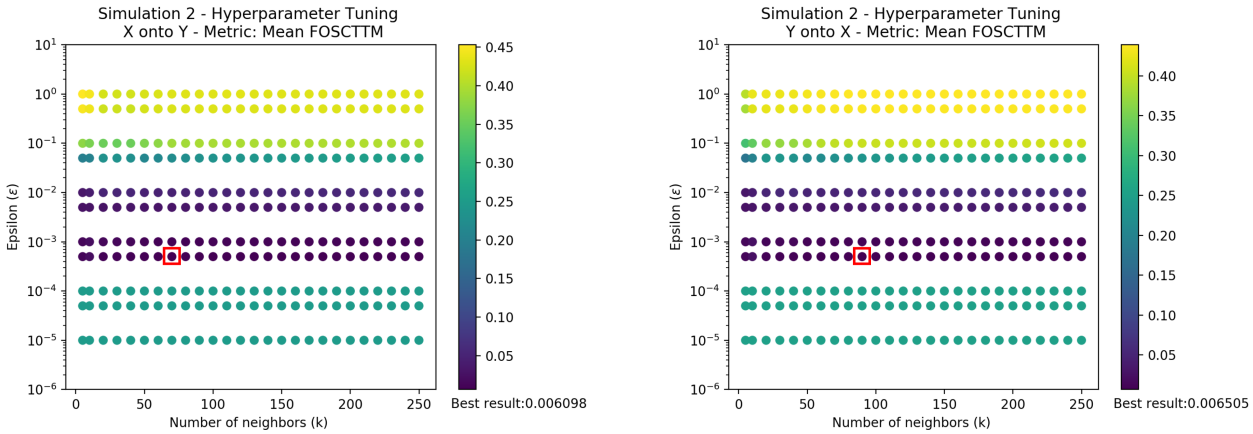
Figure S2: **Label transfer accuracy across alignment methods for scGEM data set (a)** Label transfer accuracy with varying values for k when kNN classifier is trained on the gene expression data set and predicts the cell labels of DNA methylation data set. **(b)** Label transfer accuracy with varying values for k when kNN classifier is trained on the DNA methylation data set and predicts the cell labels of gene expression data set.

1.4 Hyperparameter Tuning for SCOT



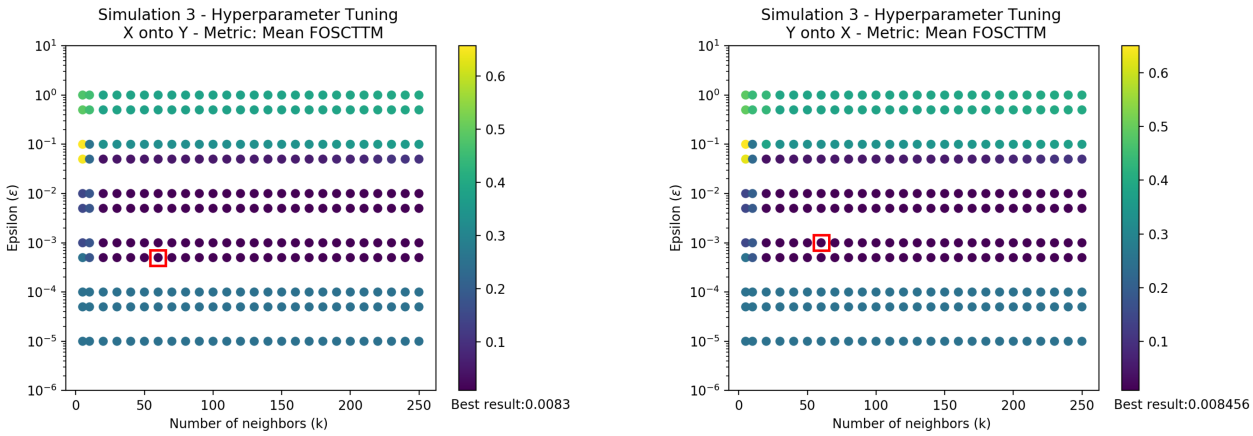
(a) (b)

Figure S3: **Hyperparameter optimization results for simulation data set 1.** Mean FOSCTTM metric was used to assess performance. **(a)** Results when domain 1 (X) is projected onto domain 2 (y). **(b)** Results when domain 2 (y) is projected onto domain 1 (X).The algorithm is largely robust to the choice of k. For domain 1 projection on domain 2, the best performing hyperparameter setting was $\epsilon = 0.001, k = 30$. For domain 2 projection on domain 1, it was $\epsilon = 0.005, k = 80$. The hyperparameter combination that yielded the best performance is highlighted with red square. For ease of visualization, a subset of the ϵ values are plotted.



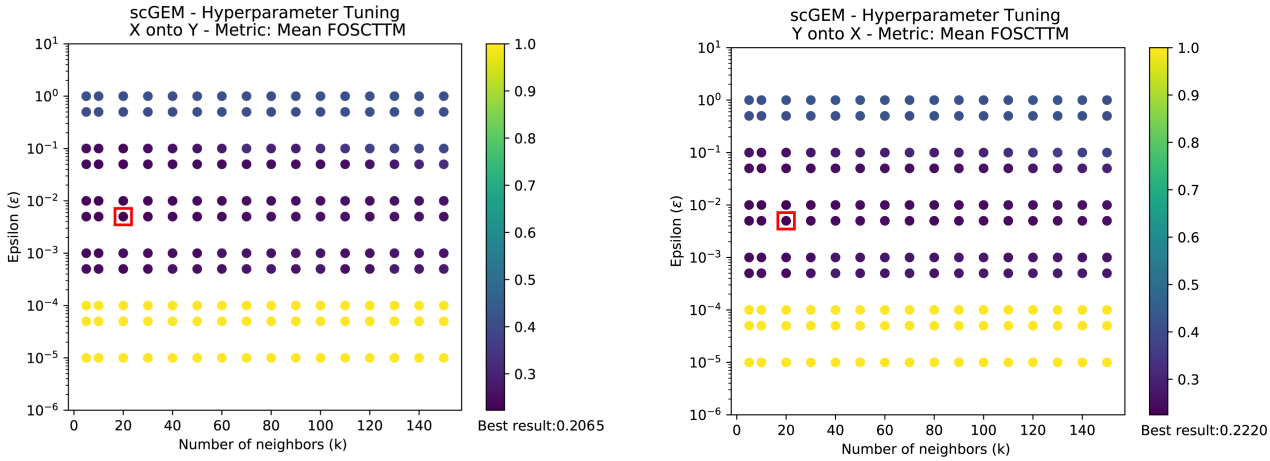
(a) (b)

Figure S4: **Hyperparameter optimization results for simulation data set 2.** Mean FOSCTTM metric was used to assess performance (indicated by color). **(a)** Results when domain 1 (X) is projected onto domain 2 (y). **(b)** Results when domain 2 (y) is projected onto domain 1 (X). The algorithm is largely robust to the choice of k. For domain 1 projection on domain 2, the best performing hyperparameter setting was $\epsilon = 0.0005, k = 70$. For domain 2 projection on domain 1, it was $\epsilon = 0.001, k = 90$. The hyperparameter combination that yielded the best performance is highlighted with red square. For ease of visualization, a subset of the ϵ values are plotted.



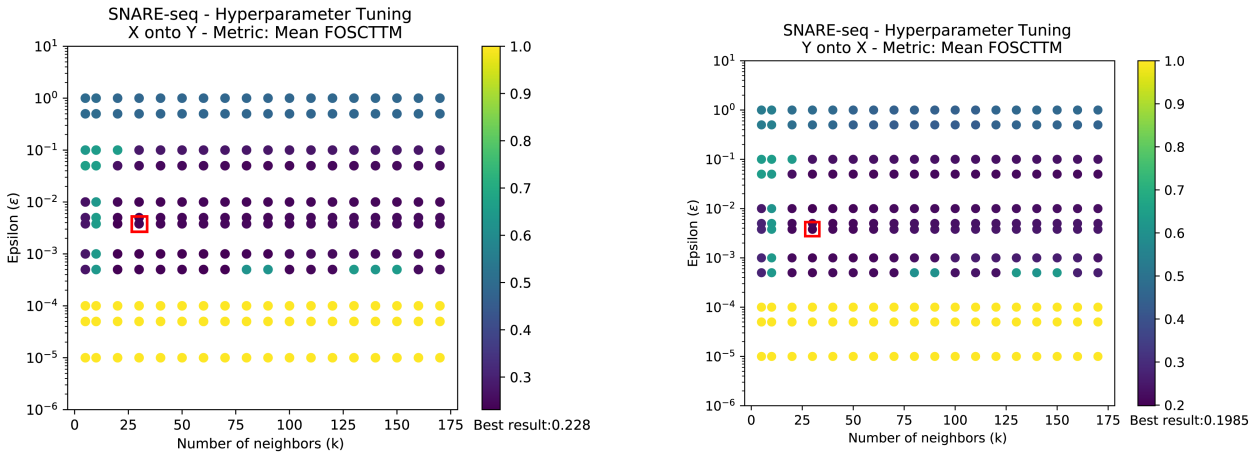
(a) (b)

Figure S5: **Hyperparameter optimization results for simulation data set 3.** Mean FOSCTTM metric was used to assess performance (indicated by color). **(a)** Results when domain 1 (X) is projected onto domain 2 (y). **(b)** Results when domain 2 (y) is projected onto domain 1 (X). The algorithm is largely robust to the choice of k. For domain 1 projection on domain 2, the best performing hyperparameter setting was $\epsilon = 0.0005, k = 60$. For domain 2 projection on domain 1, it was $\epsilon = 0.001, k = 60$. The hyperparameter combination that yielded the best performance is highlighted with red square. For ease of visualization, a subset of the ϵ values are plotted.



(a) (b)

Figure S6: **Hyperparameter optimization results for scGEM dataset.** Mean FOSCTTM metric was used to assess performance (indicated by color). **(a)** Results when gene expression domain (X) is projected onto DNA methylation domain (y). **(b)** Results when DNA methylation domain (y) is projected onto gene expression domain (X). The algorithm is largely robust to the choice of k. For both projections, the best performing hyperparameter setting was $\epsilon = 0.005, k = 20$. The hyperparameter combination that yielded the best performance is highlighted with red square. For ease of visualization, a subset of the ϵ values are plotted.



(a) (b)

Figure S7: **Hyperparameter optimization results for SNARE-seq dataset.** Mean FOSCTTM metric was used to assess performance (indicated by color). **(a)** Results when chromatin accessibility domain (X) is projected onto gene expression domain (y). **(b)** Results when expression domain (y) is projected onto chromatin accessibility domain (X). The algorithm is largely robust to the choice of k. For both projections, he best performing hyperparameter setting was $\epsilon = 0.0038, k = 30$. The hyperparameter combination that yielded the best performance is highlighted with red square. For ease of visualization, a subset of the ϵ values are plotted.