bioRxiv preprint doi: https://doi.org/10.1101/644310; this version posted May 21, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Jointly embedding multiple single-cell omics measurements

Jie Liu
Department of Computational Medicine and Bioinformatics, University of Michigan
drjieliu@umich.edu

Yuanhao Huang
Department of Computational Medicine and Bioinformatics, University of Michigan
hyhao@umich.edu

Ritambhara Singh
Department of Genome Sciences, University of Washington
rsingh7@uw.edu

Jean-Philippe Vert
Google Brain, Paris
Centre for Computational Biology, MINES ParisTech, PSL University
jpvert@google.com

William Stafford Noble
Department of Genome Sciences, University of Washington
Paul G. Allen School of Computer Science and Engineering, University of Washington
william-noble@uw.edu

May 20, 2019

Abstract

Many single-cell sequencing technologies are now available, but it is still difficult to apply multiple sequencing technologies to the same single cell. In this paper, we propose an unsupervised manifold alignment algorithm, MMD-MA, for integrating multiple measurements carried out on disjoint aliquots of a given population of cells. Effectively, MMD-MA performs an *in silico* co-assay by embedding cells measured in different ways into a learned latent space. In the MMD-MA algorithm, single-cell data points from multiple domains are aligned by optimizing an objective function with three components: (1) a maximum mean discrepancy (MMD) term to encourage the differently measured points to have similar distributions in the latent space, (2) a distortion term to preserve the structure of the data between the input space and the latent space, and (3) a penalty term to avoid collapse to a trivial solution. Notably, MMD-MA does not require any correspondence information across data modalities, either between the cells or between the features. Furthermore, MMD-MA's weak distributional requirements for the domains to be aligned allow the algorithm to integrate heterogeneous types of single cell measures, such as gene expression, DNA accessibility, chromatin organization, methylation, and imaging data. We demonstrate the utility of MMD-MA in simulation experiments and using a real data set involving single-cell gene expression and methylation data.

1 Introduction

Next-generation sequencing has enabled high-throughput interrogation of many different physical properties of the genome, including the primary DNA sequence but also the expression of messenger RNAs, localized binding of specific factors, histone modifications, nucleosome occupancy, chromatin accessibility, etc. Most of these sequencing assays have been performed on populations of cells. However, such bulk measurements

bioRxiv preprint doi: https://doi.org/10.1101/644310; this version posted May 21, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

do not allow for easy characterization of systematic or stochastic variations in physical properties of the genome among cells within a given population. Over the past several years, a variety of genomic assays have been modified to allow characterization of single cells. These modifications sometimes involve physically segregating individual cells prior to sequencing, or alternatively involve successive rounds of DNA bar-coding to identify reads derived from single cells. Examples of single-cell genomics assays include single-cell RNA-seq (scRNA-seq) for gene expression [14], single-cell ATAC-seq (scATAC-seq) for chromatin accessibility [3], single-cell Hi-C (scHi-C) for 3D genome organization [10] and single-cell methylation analysis (scMethyl-seq) [13]. In each case, the result is a data set that, compared to a standard, bulk genomic assay, has an additional dimension corresponding to the cells in the sample population.

Single-cell measurements are valuable because they permit a view of the cell-to-cell variation of a given type of physical measurement of the genome. However, such measurements would be even more valuable if multiple different measurements could be obtained from the same individual cell. Such co-assays are feasible, albeit challenging and lower throughput, for pairs of assays, such as scRNA-seq and scATAC-seq [4] or scRNA-seq and scMethyl-seq [2], that measure orthogonal physical properties. However, other pairs of single-cell assays, such as scATAC-seq and scHi-C cannot be paired even in principle, because each assay operates on (and cleaves) the genomic DNA.

In this paper, we propose a manifold alignment algorithm based on the maximum mean discrepancy (MMD) measure, called MMD-MA, which can integrate different types of single-cell measurements. Our MMD-MA algorithm assumes that the cells are drawn from the same initial population—e.g., cells of the same type or a distribution of cell types from the same experimental conditions—but the algorithm does not require any correspondence information either among samples or among the features in different domains. The algorithm makes no parametric assumptions about the forms of the distributions underlying the various measurements. The only assumption is that the distributions share a latent structure with sufficient variability that the MMD term in the optimization can align the distributions. For example, if both underlying distributions are simple isotropic Gaussian distributions, then it will not be possible to reconstruct the relative orientation of the alignment. In practice, visualization of many different single-cell data sets using dimensionality reduction methods such as PCA, t-SNE or UMAP suggest that they commonly exhibit complex structure that, we hypothesize, should allow for alignment across data modalities. In particular, MMD-MA can be applied to many heterogeneous types of single cell measures, including gene expression, DNA accessibility, chromatin organization, methylation, and imaging data. Thus, the algorithm allows us, in principle, to obtain the insights offered by a single-cell co-assay by computationally integrating two or more separate sets of single-cell measurements derived from the same or similar populations of cells. We demonstrate the performance of the algorithm on three simulated data set as well as one real data set consisting of gene expression and methylation profiles of single cells.

2 Methods

2.1 The unsupervised manifold alignment problem

Our goal is to automatically discover a manifold structure that is shared among two or more sets of points that have been measured in different ways, i.e., which mathematically live in different spaces. For simplicity, we describe the case where we have just two types of measurements, though the approach generalizes to any number of input domains. Let the two sets of points be $X^{(1)} = \left(x_1^{(1)}, x_2^{(1)}, ..., x_{n_1}^{(1)}\right)$ from $\mathcal{X}^{(1)}$ and $X^{(2)} = \left(x_1^{(2)}, x_2^{(2)}, ..., x_{n_2}^{(2)}\right)$ from $\mathcal{X}^{(2)}$. The numbers of data points in the two domains are n_1 and n_2, respectively. We do not require any correspondence information regarding the measurements across different domains or regarding the samples from different domains. Instead, we assume that both sets of points share a manifold structure, which we aim to discover in an unsupervised fashion.

To ensure the generality of our approach, we frame the optimization using kernels. Hence, we assume that we have a way of calculating similarities between pairs of entities from the same domain, using positive definite kernel functions $k_I : \mathcal{X}^{(I)} \times \mathcal{X}^{(I)} \to \mathbb{R}$ for $I = 1, 2$. The resulting kernel Gram matrices are denoted by $K_1 \in \mathbb{R}^{n_1 \times n_1}$ and $K_2 \in \mathbb{R}^{n_2 \times n_2}$, where $[K_I]_{ij} = k_I\left(x_i^{(I)}, x_j^{(I)}\right)$ for $I = 1, 2$ and $1 \leq i, j \leq n_I$. As long as both kernel functions are positive definite, then we are guaranteed that each kernel corresponds to the

bioRxiv preprint doi: https://doi.org/10.1101/644310; this version posted May 21, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

scalar product operation in some induced feature space, and that there exists a space of functions $\mathcal{H}_I$, called a reproducing kernel Hilbert space (RKHS), mapping $\mathcal{X}^{(I)}$ to $\mathbb{R}$ endowed with a Hilbert space structure. For example, if the input space $\mathcal{X}$ is a vector space and we take the linear kernel $k(x, x') = x^\top x'$, then the RKHS is made of linear functions of the form $f(x) = w^\top x$, endowed with the norm $||f|| = ||w||$. If we take a nonlinear kernel such as the Gaussian RBF kernel $k(x, x') = \exp\left(-||x - x'||^2/(2\sigma^2)\right)$ with bandwidth $\sigma > 0$, then the RKHS contains nonlinear functions. The use of kernels allows the MMD-MA algorithm to operate in principal on any type of entity—vector, graph, string, etc.— for which a kernel function can be defined.

In order to find a shared structure between the points in $\mathcal{X}^{(1)}$ and $\mathcal{X}^{(2)}$, we propose to learn two mappings $\phi_1 : \mathcal{X}^{(1)} \to \mathbb{R}^p$ and $\phi_2 : \mathcal{X}^{(2)} \to \mathbb{R}^p$, so that input data in different spaces are all mapped to the same p-dimensional space $\mathbb{R}^p$ and can be compared in that space. For each $I = 1, 2$, we consider each coordinate of ϕ_I in the RKHS $\mathcal{H}_I$ of the corresponding kernel k_I, i.e., $\phi_I = \left(\phi_1^{(I)}, \ldots, \phi_p^{(I)}\right) \in \mathcal{H}_I^p$. We then consider each function $\phi_j^{(I)} \in \mathcal{H}_I$ of the form $\phi_j^{(I)}(x) = \sum_{\ell=1}^{n_I} \alpha_{\ell j}^{(I)} k_I(x_\ell^{(I)}, x)$, for any $x \in \mathcal{X}^{(I)}$. This parametrization of $\phi_j^{(I)}$ in terms of $\alpha_{\ell j}^{(I)}$'s always exists by the representer theorem, provided we regularize the optimization problem with the RKHS norm of $\phi_j^{(I)}$, as we explain below. Now, if we denote by α_i the $n_i \times p$ matrix with entries $\alpha_{\ell j}^{(I)}$, then $K_I \alpha_I$ is the $n_I \times p$ matrix where the j-th row (for $j = 1, \ldots, n_I$) is the p-dimensional image of $x_j^{(I)}$ by the mapping ϕ_I. In addition, $\alpha_I^\top K_I \alpha_I$ is the $p \times p$ matrix of inner products in the RKHS of the p functions $\phi_1^{(I)}, \ldots, \phi_p^{(I)}$, which is for example equal to the $p \times p$ identity matrix I_p when ϕ_I is a projection onto a subspace of dimension p in the RKHS. In order to define MMD-MA, we now discuss the criteria to optimize for α_1 and α_2 in order to discover shared structures between the two views.

2.2 Characterizing the distribution distance in the shared space

Although we do not assume we know the individual correspondence between points in the two domains, or even that such a 1-to-1 mapping exists, we do assume that the two distributions of points are similar in the shared space. Thus, the optimal mapping matrices α_1 and α_2 should make the two mapped sets of points in the shared space, namely $K_1\alpha_1$ and $K_2\alpha_2$, as similar as possible. To specify the distance between the two mapped manifolds in the shared space, we use an MMD term $\mathrm{MMD}(K_1\alpha_1, K_2\alpha_2)^2$, which is a general, differentiable measure of how similar two clouds of points are [7]. Formally, MMD is defined through a positive definite kernel K_M over $\mathbb{R}^p$ through the formula

$$\mathrm{MMD}^2\left(\left\{u_1^{(1)}, \ldots, u_{n_1}^{(1)}\right\}, \left\{u_1^{(2)}, \ldots, u_{n_2}^{(2)}\right\}\right) = \frac{1}{n_1^2}\sum_{i,j=1}^{n_1} K_M(u_i^{(1)}, u_j^{(1)}) - \frac{2}{n_1 n_2}\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} K_M(u_i^{(1)}, u_j^{(2)}) + \frac{1}{n_2^2}\sum_{i,j=1}^{n_2} K_M(u_i^{(2)}, u_j^{(2)}),$$

where we denote $u_i^{(I)} = \phi_I(x_i^{(I)})$ to simplify notation. In this work, we use a Gaussian RBF kernel for K_M, where the bandwidth parameter σ is a user-specified parameter.

Chwialkowski et al. [5] propose two fast methods (with complexity linear in $n_1 + n_2$) to estimate MMD^2, both of which are differentiable with respect to the positions of the points. Because the MMD is small when the distributions are similar, MMD-MA aims to minimize MMD^2 with respect to the embeddings.

2.3 The MMD-MA algorithm

Unfortunately, simply minimizing MMD between the two kernels is insufficient. Most notably, we need to ensure that the relationships among data points in the input space is preserved to some extent in the feature space; otherwise, the method may learn very complicated mappings that completely modify the relative positions of cells in order to have them aligned between the two views. For that purpose we introduce a distortion term $\mathrm{dis}(\alpha_I) = ||K_I - K_I\alpha_I\alpha_I^\top K_I^\top||_2$, which quantifies how the matrix of inner products between points in the original space (quantified by the kernel matrix K_i) differs from the matrix of inner products after mapping in the p-dimensional space. Penalizing $\mathrm{dis}(\alpha_I)$ ensures that the distortion between the data in the original space and the data mapped to the low-dimensional space should be small. In addition, we may

bioRxiv preprint doi: https://doi.org/10.1101/644310; this version posted May 21, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

wish to ensure that the mappings to $\mathbb{R}^p$ are (almost) projections from the high-dimensional RKHS, which we obtain by adding a penalty term $\text{pen}(\alpha_I) = ||\alpha_I^\top K_I^\top \alpha - I_p||_2$.

Thus, MMD-MA optimizes, with respect to α_1 and α_2, the following objective function:

$$\min_{\alpha_1,\alpha_2} \text{MMD}(K_1\alpha_1, K_2\alpha_2)^2 + \lambda_1(\text{pen}(\alpha_1) + \text{pen}(\alpha_2)) + \lambda_2(\text{dis}(\alpha_1) + \text{dis}(\alpha_2)) \,. \tag{1}$$

where

$$\text{pen}(\alpha_I) = ||\alpha_I^\top K_I^\top \alpha - I_p||_2^2 \,, \tag{2}$$

and

$$\text{dis}(\alpha_I) = ||K_I - K_I\alpha_I\alpha_I^\top K_I^\top||_2^2. \tag{3}$$

2.4 Solving the optimization problem

To find a stationary point of (1) we use a simple gradient descent scheme, and use Adam [9] to adjust the learning rate. Since the optimization problem is not convex, only a local minimum can be expected. Therefore, for a given data set we run the optimization procedure with 100 different random seed values and keep the solution which provides the lowest objective function value.

In practice, solving the optimization requires specifying several hyperparameters. These include the dimensionality p of the shared space, any parameters of the kernel functions K_1, K_2 and K_M, and the tradeoff parameters λ_1 and λ_2. In this work, we assume that p and the kernel parameters are user-specified, and we investigate the performance of the algorithm as we vary λ_1 and λ_2.

3 Related work

We are aware of three other methods that address the unsupervised manifold alignment problem, which we briefly review here.

3.1 The joint Laplacian manifold alignment (JLMA) algorithm

The joint Laplacian manifold alignment (JLMA) algorithm [15] performs manifold alignment by constructing a joint Laplacian across multiple domains and then performing eigenvalue decomposition to find the optimal solution. The joint Laplacian formulation can also be interpreted as preserving similarities within each view and correspondence information about instances across views, which is captured by the joint Laplacian matrix. The loss function is $C(F) = \sum_{i,j} ||\mathbf{F}(i,.) - \mathbf{F}(j,.)||^2\mathbf{W}(i,j)$, where the summation is over all pairs of instances from all views. $\mathbf{F}$ is the unified representation of all instances, and the output of the algorithm is the joint adjacency matrix $\mathbf{W}$. To avoid trivial solutions (i.e., mapping all instances to zero), JLMA includes a constraint $\mathbf{F}'\mathbf{D}\mathbf{F} = I$ where I is an identity matrix. Let $\mathbf{F} = [f_1, f_2, \ldots, f_d]$. The optimization problem then becomes

$$\underset{\mathbf{F}:\mathbf{F}'\mathbf{D}\mathbf{F}=1}{\text{argmin}} \; C(\mathbf{F}) = \underset{f_1,\ldots,f_d}{\text{argmin}} \sum_i f_i'\mathbf{L}f_i + \lambda_i(1 - f_i'\mathbf{D}f_i). \tag{4}$$

The optimal solution is the d smallest nonzero eigenvectors from the generalized eigen decomposition problem.

JLMA can be used in an unsupervised or supervised fashion. In supervised mode, the Laplacian L is given as input. In the unsupervised setting, the key step is to construct the cross-domain Laplacian submatrix of $\mathbf{L}$. Wang *et al.* use k-NN graphs to characterize local geometry and use the minimum distances from scaled permutated k-NN graphs to construct a cross-view Laplacian submatrix of $\mathbf{L}$. Thereafter, the rest of the algorithm is the same as the standard JLMA algorithm. Unfortunately, the computational cost of this initial step is quite high, even for small k values. To deal with this problem, Pe *et al.* use a B-spline curve to fit the local geometry and calculate cross-view matching scores from the curves [11]. Thus, in both cases, unsupervised manifold alignment is done via two steps: computing a cross-domain matching score, and identifying the correspondence via Equation 4.

bioRxiv preprint doi: https://doi.org/10.1101/644310; this version posted May 21, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

3.2 The generalized unsupervised manifold alignment (GUMA) algorithm

The generalized unsupervised manifold alignment (GUMA) algorithm [6] is another method that does not require any correspondence information a priori. The approach assumes that instances in the two domains (e.g., in our case, two cells measured using different techniques) can be matched to one another in a one-to-one fashion. In particular, the algorithm formulates an optimization problem whose objective function includes a geometry matching term E_s across different domains, a feature matching term E_f, and a geometry preserving term E_p, subject to a 0-1 correspondence matrix $\mathbf{F}$ and feature projections $\mathbf{P}_i$ for each domain (i.e. domain i). The optimization is performed using alternating minimization, alternating between optimizing $\mathbf{F}$ and $\mathbf{P}_i$ with the other fixed. The algorithm outputs both the instance correspondence between the domains and the feature mapping functions between the domains.

3.3 The manifold alignment generalized adversarial network (MAGAN) algorithm

MAGAN [1] consists of two GANs that learn reciprocal mappings between two domains; i.e., GAN1 learns the mapping from domain 1 to domain 2, and GAN2 learns the mapping from domain 2 to domain 1. Each GAN's generator takes input in one domain and outputs in the other domain, with the hope that the discriminator in the other domain cannot distinguish the fake samples from true samples. The loss function of the generators consists of three terms. The reconstruction loss term L_r captures the difference between a sample and itself after being mapped to the different domain and then mapped back to the original domain. The discriminator loss term L_d makes sure that the mapped sample in the other domain has a high likelihood to fool the discriminator in that domain. And the correspondence loss L_c forces the learned mapping to agree with some prior correspondence, either "unsupervised correspondence" (e.g., some variables are shared between two domains) or "semi-supervised correspondence" (e.g., some labeled pairs cross two domains). The paper empirically demonstrates that the inclusion of correspondence information greatly improves the performance of the manifold alignment.

3.4 Comparison of these three algorithms with our algorithm

The MMD term in our formulation only ensures that the two distributions agree globally in the latent space, whereas both JLMA and GUMA have a term that ensures, for each instance, that the local geometry is preserved between domains. This is the difference between *manifold superimposing* [18, 17, 8] and *manifold alignment* discussed in the MAGAN paper [1]. Furthermore, GUMA's assumption that individual cells can be matched 1-to-1 between the two input domains is not generally true, most obviously when $n_1 \neq n_2$. MAGAN [1] itself does not include a component for identifying a correspondence in an unsupervised fashion, and empirical results from the MAGAN paper suggest that the algorithm may not be useful if there is no known correspondence information between the two domains. When more than three domains must be aligned, the JLMA, GUMA, and our MMD-MA algorithms can be easily extended, whereas the formulation of MAGAN makes such an extension difficult.

4 Results

4.1 Three simulations

To validate the performance of MMD-MA, we generated three simulated data sets, each from a different d-dimensional manifold. The first manifold exhibits a branching structure in two-dimensional space (i.e., $d = 2$) to mimic a branching differentiation situation (first column of Figure 1). The second manifold structure is a nonlinear mapping of the first structure. The branching structure is mapped onto a Swiss roll manifold (second column of Figure 1). Samples in the first domain are mapped from the 2D space of simulation 1 into 3D space such that the three dimensions are $[x_1 cos(3x_1), x_2, x_1 sin(3x_1)]$, while samples in the second domain are mapped into 3D space by $[x_1 sin(2x_1), x_2, x_1 cos(2x_1)]$. The third manifold is a circular frustum in three-dimensional space (i.e., $d = 3$), which aims to mimic the cell cycle superimposed

bioRxiv preprint doi: https://doi.org/10.1101/644310; this version posted May 21, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

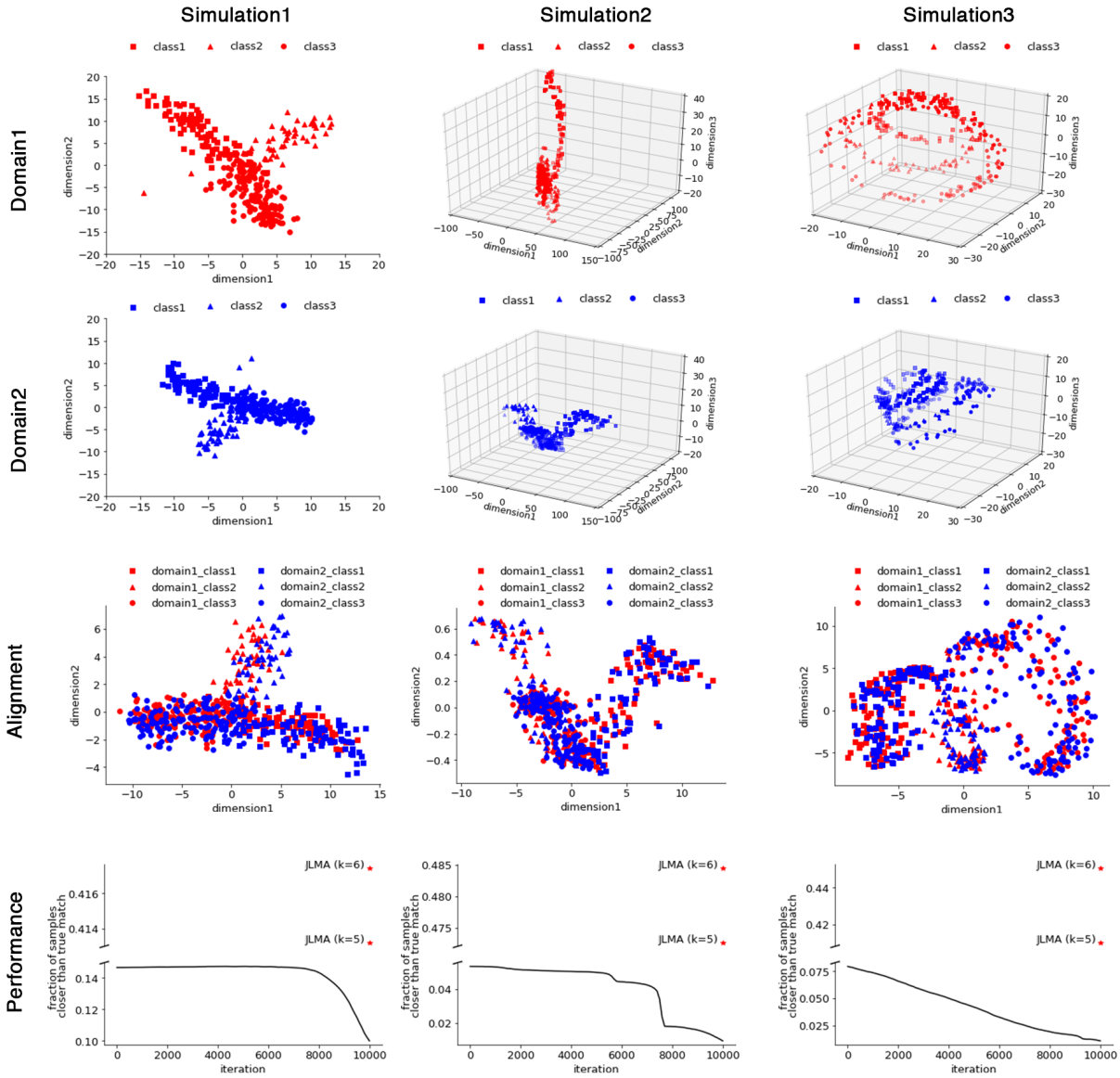

Figure 1: **Three simulation experiments.** The first two rows show the MDS projection of the data points in domain 1 and domain 2, separately. The third row shows the projection of the data points in the shared embedding space. The last row plots the fraction of samples closer than the true match as MMD-MA iterates. Points are included from JLMA when k=5 and k=6.

bioRxiv preprint doi: https://doi.org/10.1101/644310; this version posted May 21, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Table 1: **Running time of MMD-MA and JLMA.** Times are provided for the three simulation experiments and the real single cell application.

	Sim 1	Sim 2	Sim 3	Methyl-Expr
MMD-MA	0:53	0:58	0:59	0:10
JLMA, $k = 5$	4:06	4:07	4:20	1:27
JLMA, $k = 6$	26:11	25:56	32:16	2:16

on a linear differentiation process (third column of Figure 1). From each of these d-dimensional manifolds, we simulated $n = 300$ data points, and we refer to the corresponding $n \times d$ data matrix as Z.

For each manifold, we assume that we have two domains, and we generate a $d \times p_1$ mapping matrix T_1 and a $d \times p_2$ mapping matrix T_2. Each element in the two mapping matrices is sampled from a standard Gaussian distribution. We set the observed data matrix from domain 1 to be ZT_1 and the observed data matrix from domain 2 to be ZT_2. For example, we set $p_1 = 1000$ and $p_2 = 2000$. We also add Gaussian noise ($\sigma = 0.05$) to each element of the covariates. For all of the simulations, we set the kernel matrices K_1 and K_2 to be the inner product of the $z-$normalized observed data matrices. For each simulation, we intentionally mis-specify, as input to MMD-MA, the dimensionality of the latent space as $p = 5$, to simulate the scenario in which the true latent dimensionality is unknown a priori.

For all three numerical simulations, we plot the data points in the projected space, namely $K_1\alpha_1^{(o)}$ and $K_2\alpha_2^{(o)}$ (Figure 1). In all three cases, the two domains appear to be aligned correctly in the latent space. To quantify this alignment, we use the known correspondence between points in the two domains as follows. For each point x in one domain, we identify its (true) nearest neighbor in the other domain. We then rank all data points in the learned latent space by their distance from x, and we compute the fraction of points that are closer than the true nearest neighbor. Averaging this fraction across all data points in both domains yields the "average fraction of samples closer than the true match," where perfect recovery of the true manifold structure yields values close to zero. In all three simulations, the observed fraction of samples closer than the true match decreases monotonically and approaches zero as the MMD-MA algorithm iterates.

Finally, we attempted to compare the performance of other algorithms on the same simulated data sets. Unfortunately, we had difficulty running the GUMA algorithm using the implementation shared by the authors; hence, we leave GUMA out of our comparison. Similarly, we could not include the MAGAN algorithm because it requires some initial correspondence information, which we are assuming is not available. Consequently, we only compare to the JLMA algorithm as a baseline, using $k = 5$ (the default value) and $k = 6$. We find that, in all three simulations, our MMD-MA algorithm outperforms the baseline JLMA (bottom row of Figure 1).

The running time of MMD-MA is much lower than JLMA using either $k = 5$ or $k = 6$. Timings on an Intel Xeon Gold 6136 CPU at 3.00GHz (Table 1) show that MMD-MA runs under one minute, considerably faster than JLMA.

4.2 Real world application results

In a recent study, gene expression levels and methylation rates were profiled jointly in 61 single cells [2]. We use this co-assay data to validate our method by hiding the correspondence between genes from MMD-MA and then measuring how well the correspondence is recovered. Prior to analysis, we remove the genes where any of the cells have a missing value for either the methylation rate or gene expression. This step leaves, for the 61 cells, 2486 genes with both methylation rate and gene expression measured. We regard gene expression as domain 1 and methylation rate as domain 2. We pretend that we do not know the correspondence information, run our MMD-MA algorithm, and see how well our algorithm can align the two manifolds and recover the cell correspondence. For calculating the similarity kernel matrices K_1 and K_2, we first perform z-score normalization on the gene expression levels and the methylation rates and then calculate the inner product for the elements in the cell-by-cell similarity matrices. As in the simulations, we embed the two domains into a latent space of dimensionality $p = 5$.

We first plot the Principal Component Analysis (PCA) projection of the single cells based on their gene expression levels and their methylation rates separately (Figure 2A). In this plot, when we connect the two

bioRxiv preprint doi: https://doi.org/10.1101/644310; this version posted May 21, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

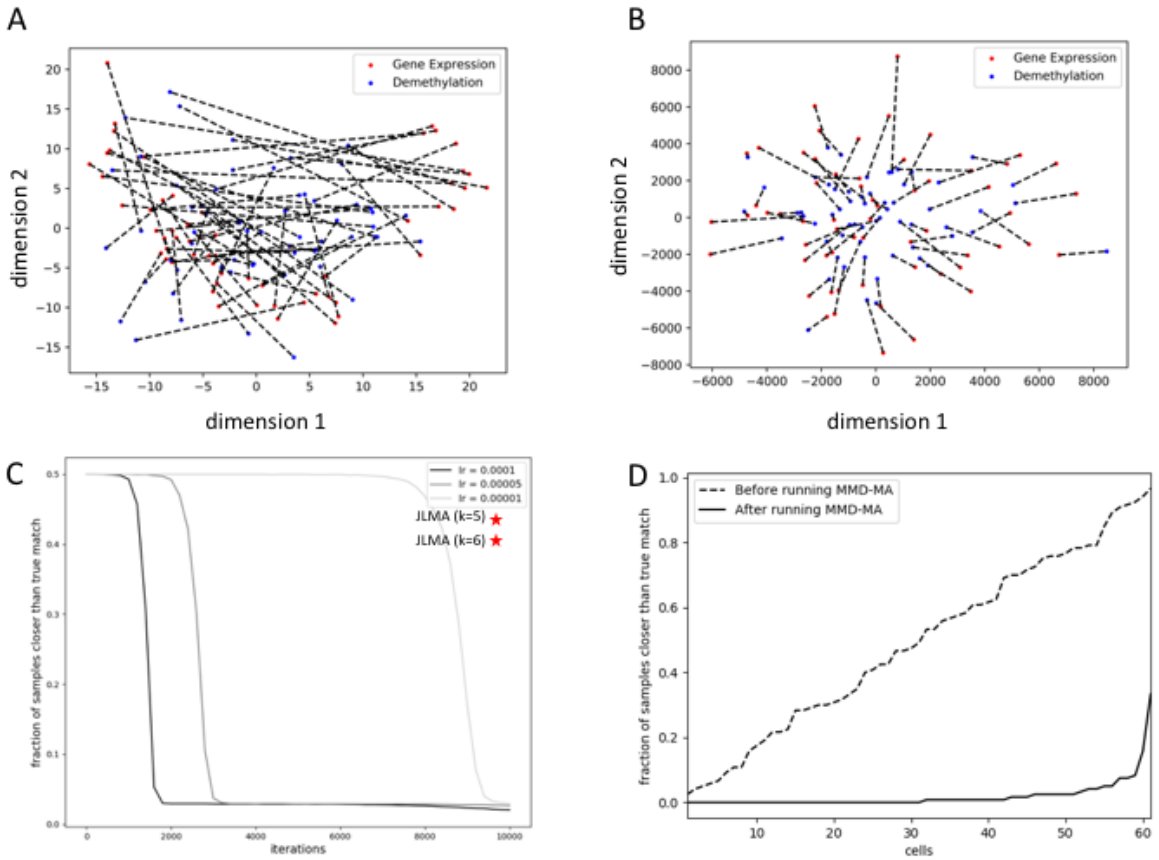

Figure 2: **Results from real world single cell applications.** (A) PCA projection of single cells based on their gene expression levels and their methylation rates separately, with dotted lines connecting the same cell. (B) Projection of the single cells in the shared space from the MMD-MA algorithm, with dotted lines connecting the same cell. (C) The average fraction of samples closer than the true match decreases as MMD-MA iterates. This result is consistent across different learning rates of the optimization. (D) The fraction of samples closer to each cell than its true match is plotted before and after MMD-MA, with the 61 cells in sorted order along the x-axis. For each cell, the average is computed separately for each domain, and then the two values are averaged together. The fraction is high and close to uniformly distributed before running the MMD-MA algorithm and reduces considerably as the algorithm learns the aligned shared space.

bioRxiv preprint doi: https://doi.org/10.1101/644310; this version posted May 21, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Table 2: **Properties and hyperparameters of the experiments.**

	Sim 1	Sim 2	Sim 3	Methyl-Expr
number of Samples	300	300	300	61
dimension (Domain 1)	1000	1000	1000	2486
dimension (Domain 2)	2000	2000	2000	2486
λ_1	1e-6	1e-9	1e-5	1e-2
λ_2	1e-2	1e-7	1e-6	1e-6
σ	0.5	0.1	1.2	10000

dots corresponding to the same cell, we observe that each cell tends to be projected to two different locations in the latent space. Accordingly, the average fraction of data points closer than the true match is 0.49. We then run MMD-MA algorithm on this dataset and plot the PCA projection of the 61 single cells in terms of gene expression and methylation rate in the shared space recovered by MMD-MA (Figure 2B). In the shared space projection, we connect the two embeddings from different perspectives, and we observe that the cells are embedded well in the shared space. Next, we calculate the fraction of samples closer to each cell that its true match in the shared space of dimensionality $p = 5$. This fraction decreases as MMD-MA iterates, reaching 0.024 in the end, and the trend is consistent across different learning rates of the optimization (Figure 2C). An alternative visualization of the per-cell fractions before and after optimization (Figure 2D) further illustrates that the MMD-MA algorithm successfully maps >50% of the cells closest to their true neighbor in the other domain.

4.3 MMD-MA's performance is robust to variations in hyperparameters

Running the MMD-MA algorithm requires specifying several hyperparameters. We investigated the robustness of the learned embedding relative to variations in these hyperparameters.

As noted previously, in all of our studies the dimensionality p of the latent space has been set to 5 even though the correct number should be $p = 2$ in the first two simulations, $p = 3$ in the third simulation, and is unknown for the Methyl-Expr data set. We observe that MMD-MA algorithm can still align the two manifolds even when the dimensionality parameter p is misspecified.

The trade-off parameters λ_1 and λ_2 determine how much the three terms contribute to the overall objective function. In this work, we set these trade-off parameters by monitoring whether the three terms have comparable magnitudes or whether one particular term dominates in the converged solution. We tested eight combinations of these trade-off parameters for each data set (Supplementary Table S1). In each case, we observe that the performance of MMD-MA is almost the same with different choices of trade-off parameters, although some trade-off parameters may lead to a different convergence path (Supplementary Figure S1 A–D).

The bandwidth parameter σ associated with the Gaussian kernel K_M in the MMD term determines how much each data point contributes to its neighborhood in the calculation of the MMD. The σ values used in our experiments are shown in Table 2. We also tested different values of σ and observed that the performance of MMD-MA is quite invariant to them (Supplementary Figure S1 E–H).

5 Discussion

In this paper, we propose an unsupervised manifold alignment algorithm, MMD-MA, for integrating multiple types of single-cell measurements carried out on disjoint populations of single cells drawn from a common source. The key advantage of our MMD-MA algorithm is that it does not require any correspondence information, either between the samples or between the features. In many real-world integration applications, such correspondence information is not available. Another advantage of our MMD-MA algorithm is that has only weak distributional requirements for the domains to be aligned, namely, that the manifolds exhibit sufficient structure to allow for alignment. This flexibility gives MMD-MA the power to potentially integrate many different types of single cell measures, including gene expression, DNA accessibility, chromatin organization, methylation, and imaging data. Furthermore, the MMD-MA framework can easily be extended to

bioRxiv preprint doi: https://doi.org/10.1101/644310; this version posted May 21, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

more than two domains, allowing integration of, for example, scRNA-seq, scATAC-seq, and scHi-C of single cells. We have shown that MMD-MA works well in the presence of nonlinear mappings and is robust to the choice of several hyperparameters, including the trade-off parameters, the parameters associated with the MMD term, and the dimensionality of the shared space.

Currently, MMD-MA can be used to align hundreds or thousands of single cells in a reasonable running time. The gradient descent algorithm could be parallelized to save time if multiple cores are available. For future work, we will focus on scaling up the MMD-MA algorithm. Given decreasing sequencing costs, it is likely we will need to apply MMD-MA to align millions of single cells in the future. Running MMD-MA efficiently without storing large kernel matrices in memory will be a crucial issue to solve. A possible solution may rely on random projection [12] or Nystrom approximation [16], which are approximation methods for large-scale kernel matrices.

All of the data sets used in this study, including the original and mapped simulated data and the Methyl-Expr data set, as well as the corresponding MMD-MA outputs, are available for download from `http://noble.gs.washington.edu/proj/mmd-ma`.

Acknowledgments: This work was supported by National Institutes of Health award U54 DK107979.

References

[1] M. Amodio and S. Krishnaswamy. MAGAN: Aligning biological manifolds. In *Proceedings of the International Conference on Machine Learning*, 2018.

[2] C. Angermueller, S. J. Clark, H. J. Lee, I. C. Macaulay, M. J. Teng, T. X. Hu, F. Krueger, S. A. Smallwood, C. P. Ponting, T. Voet, G. Kelsey, O.r Stegle, and W. Reik. Parallel single-cell sequencing links transcriptional and epigenetic heterogeneity. *Nature Methods*, 13:229–232, 2016.

[3] J. D. Buenrostro, B. Wu, U. M. Litzenburger, D. Ruff, M. L. Gonzales, M. P. Snyder, H. Y. Chang, and W. Greenleaf. Single-cell chromatin accessibility reveals principles of regulatory variation. *Nature*, 523(7561):486–490, 2015.

[4] J. Cao, D. A. Cusanovich, V. Ramani, D. Aghamirzaie, H. A. Pliner, A. J. Hill, R. M. Daza, J. L. McFaline-Figueroa, J. S. Packer, L. Christiansen, F. J. Steemers, A. C. Adey, C. Trapnell, and J. Shendure. Joint profiling of chromatin accessibility and gene expression in thousands of single cells. *Science*, 361(6409):1380–1385, 2018.

[5] K. Chwialkowski, A. Ramdas, D. Sejdinovic, and A. Gretton. Fast two-sample testing with analytic representations of probability measures. In *Advances in Neural Information Processing Systems*, pages 1981–1989, 2015.

[6] Z. Cui, H. Chang, S. Shan, and X. Chen. Generalized unsupervised manifold alignment. In *Advances in Neural Information Processing Systems*, pages 2429–2437, 2014.

[7] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012.

[8] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim. Learning to discover cross-domain relations with generative adversarial networks. arXiv:1703.05192, 2017.

[9] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

[10] T. Nagano, Y. Lubling, T. J. Stevens, S. Schoenfelder, E. Yaffe, W. Dean, E. D. Laue, A. Tanay, and P. Fraser. Single-cell Hi-C reveals cell-to-cell variability in chromosome structure. *Nature*, 502(7469):59–64, 2013.

[11] Y. Pei, F. Huang, F. Shi, and H. Zha. Unsupervised image matching based on manifold alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(8):1658–1664, 2012.

bioRxiv preprint doi: https://doi.org/10.1101/644310; this version posted May 21, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

[12] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, 2007.

[13] S. A. Smallwood, H. J. Lee, C. Angermueller, F. Krueger, H. Saadeh, J. Peat, S. R. Andrews, O. Stegle, W. Reik, and G. Kelsey. Single-cell genome-wide bisulfite sequencing for assessing epigenetic heterogeneity. *Nature Methods*, 11:817–820, 2014.

[14] F. Tang, C. Barbacioru, Y. Wang, E. Nordman, C. Lee, N. Xu, X. Wang, J. Bodeau, B. B. Tuch, A. Siddiqui, K. Lao, and M. A. Surani. mRNA-Seq whole-transcriptome analysis of a single cell. *Nature Methods*, 6:377–382, 2009.

[15] C. Wang, P. Krafft, and S. Mahadevan. Manifold alignment. In Y. Ma and Y. Fu, editors, *Manifold Learning: Theory and Applications*. CRC Press, 2011.

[16] C. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems*, 2001.

[17] Z. Yi, H. Zhang, P. Tan, and M. Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *ICCV*, pages 2868–2876, 2017.

[18] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. arXiv:1703.10593, 2017.

bioRxiv preprint doi: https://doi.org/10.1101/644310; this version posted May 21, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Supplement

Table S1: **Trade-off hyperparameters investigated for each data set.**

Sim	λ_1	λ_2
Sim 1	1e-4	1
Sim 1	1e-4	1e-2
Sim 1	1e-4	1e-4
Sim 1	1e-6	1
Sim 1	1e-6	1e-4
Sim 1	1e-8	1
Sim 1	1e-8	1e-2
Sim 1	1e-8	1e-4
Sim 2	1e-7	1e-5
Sim 2	1e-7	1e-7
Sim 2	1e-7	1e-9
Sim 2	1e-9	1e-5
Sim 2	1e-9	1e-9
Sim 2	1e-11	1e-5
Sim 2	1e-11	1e-7
Sim 2	1e-11	1e-9
Sim 3	1e-3	1e-4
Sim 3	1e-3	1e-6
Sim 3	1e-3	1e-8
Sim 3	1e-5	1e-4
Sim 3	1e-5	1e-8
Sim 3	1e-7	1e-4
Sim 3	1e-7	1e-6
Sim 3	1e-7	1e-8
Methyl-Expr	1	1e-4
Methyl-Expr	1	1e-6
Methyl-Expr	1	1e-8
Methyl-Expr	1e-2	1e-4
Methyl-Expr	1e-2	1e-8
Methyl-Expr	1e-4	1e-4
Methyl-Expr	1e-4	1e-6
Methyl-Expr	1e-4	1e-8

bioRxiv preprint doi: https://doi.org/10.1101/644310; this version posted May 21, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

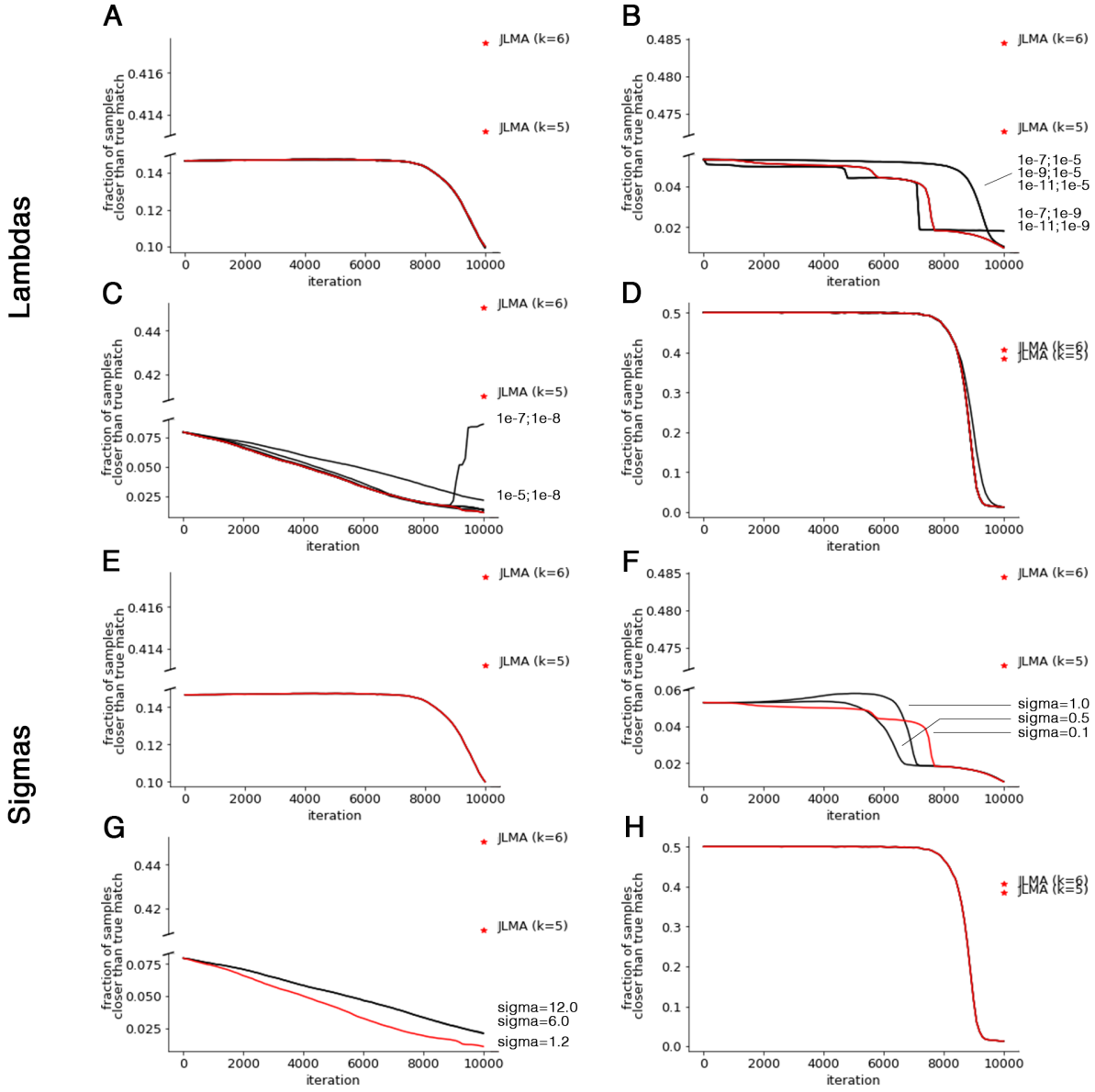

Figure S1: **Testing sensitivity to hyperparameters.** (A–D) The performance of MMD-MA when the trade-off parameters λ_1 and λ_2 are set differently in the three numerical simulations and the one real-world application, respectively. In each case, eight settings we chosen, and the plot only shows curves that differ from the one produced by the selected parameters. (E–H) The performance of MMD-MA when σ is set differently in the three numerical simulations and the one real-world application, respectively. Again, only settings that yield results different from the selected results are shown.