Spectral clustering of single-cell multi-omics data on multilayer graphs

Shuyi Zhang[1,2,†], Jacob R. Leistico [1,2,†], Raymond J. Cho[3], Jeffrey B. Cheng [3] and Jun S. Song [1,2,*]

[1]Department of Physics, University of Illinois at Urbana-Champaign, Urbana, IL
[2]Carl R. Woese Institute for Genomic Biology, University of Illinois at Urbana-Champaign, Urbana, IL
[3]Department of Dermatology, University of California, San Francisco, San Francisco, CA

[†] Co-first authors
[*] Correspondence: songj@illinois.edu

Abstract

Single-cell sequencing technologies that simultaneously generate multimodal cellular profiles present opportunities for improved understanding of cell heterogeneity in tissues. How the multimodal information can be integrated to obtain a common cell type identification, however, poses a computational challenge. Multilayer graphs provide a natural representation of multi-omic single-cell sequencing datasets, and finding cell clusters may be understood as a multilayer graph partition problem.

We introduce two spectral algorithms on multilayer graphs, spectral clustering on multilayer graphs (SCML) and the weighted locally linear (WLL) method, to cluster cells in multi-omic single-cell sequencing datasets. We connect these algorithms through a unifying mathematical framework that represents each layer using a Hamiltonian operator and a mixture of its eigenstates to integrate the multiple graph layers, demonstrating in the process that the WLL method is a rigorous multilayer spectral graph theoretic reformulation of the popular Seurat weighted nearest neighbor (WNN) algorithm. Implementing our algorithms and applying them to a CITE-seq dataset of cord blood mononuclear cells yields results similar to the Seurat WNN analysis. Our work thus extends spectral methods to multimodal single-cell data analysis.

The code used in this study can be found at https://github.com/jssong-lab/sc-spectrum

1 Introduction

Graph-based data analysis and machine learning methods have been successfully used for compressed representation of data and clustering (Ng *et al.*, 2001; Belkin and Niyogi, 2003; von Luxburg, 2007; Dong *et al.*, 2013; McInnes *et al.*, 2018b; Zhao and Song, 2018; El Gheche *et al.*, 2020). The mathematical structure of graphs greatly facilitates the analysis of real-world networks such as computer, gene regulatory, protein-protein interaction, and social networks. Graph-based methods are particularly well suited for exploring high-dimensional data in single-cell genomics by succinctly capturing the results of profiling genome-wide transcripts, chromatin accessibility, or surface proteins in tens of thousands of individual cells via next-generation sequencing (NGS)-based technologies. By constructing a similarity graph, with each node representing a cell and each link between two nodes representing their similarity, information about the neighborhood of cells on the graph can be used to capture distinct cell types reflected in the clustering structure of nodes. Using pairwise similarities between data points can also help reduce the dimension of data; e.g., t-SNE (t-Distributed Stochastic Neighbor Embedding) (van der Maaten and Hinton, 2008) and UMAP (Uniform Manifold Approximation and Projection) (McInnes *et al.*, 2018b) are two nonlinear dimension reduction methods based on pairwise similarities and have been extensively used for visualizing single-cell data.

Recent technological advances have enabled simultaneous generation of multi-omic data in single cells. For example, technologies like cellular indexing of transcriptomes and epitopes by sequencing (CITE-seq) (Stoeckius *et al.*, 2017; Liu *et al.*, 2020), as well as RNA expression and protein sequencing (REAP-seq) (Peterson *et al.*, 2017) and AbSeq (Shahi *et al.*, 2017), have made it possible to simultaneously perform single-cell RNA profiling and immunophenotyping (via profiling surface proteins). The rapid generation of single-cell multi-omic data has stimulated interest in developing graph-based algorithms that can integrate multimodal cellular profiles for dimensionality reduction and improved cell type identification. For example, CiteFuse (Kim *et al.*, 2020) integrates the transcriptome and epitope tagging information of CITE-seq by using a similarity network fusion algorithm (Wang *et al.*, 2014). The t-SNE and UMAP algorithms have been also generalized to handle single-cell multimodal data (Canzar *et al.*, 2021). More recently, the weighted nearest neighbor (WNN) algorithm, built on top of the popular Seurat workflow for analyzing individual assays (Stuart *et al.*, 2019), utilizes a weighted summation of the resulting multiple graphs to obtain a joint-similarity WNN graph (Hao *et al.*, 2021); the key idea is to perform within-modality and cross-modality predictions and to assign cell-specific modularity weights based on these predictions to construct the WNN graph.

Spectral clustering is currently one of the state-of-the-art algorithms for graph partitioning and can also facilitate dimension reduction. This algorithm does not make strong

assumptions about the shape of clusters in the original point cloud and can be very fast for analyzing large sparse graphs (Ng *et al.*, 2001; von Luxburg, 2007). While different variations exist, spectral clustering typically involves constructing a second-order difference operator called the graph Laplacian and using its first few lowest-frequency eigenvectors to embed and cluster the graph nodes in a low-dimensional space (Roweis and Saul, 2000; Ng *et al.*, 2001; Belkin and Niyogi, 2003; Dhillon *et al.*, 2007). Spectral analysis of graphs is incorporated into many algorithms and workflows. For example, the recently developed UMAP algorithm for dimension reduction (McInnes *et al.*, 2018b) is built upon the same mathematical foundations as the previous Laplacian Eigenmaps (Belkin and Niyogi, 2003). CiteFuse also uses spectral clustering to partition the fused graph and has shown that it outperforms alternative methods (Kim *et al.*, 2020).

A multilayer graph is a set of graphs defined on the same nodes and can simultaneously represent several multi-omic single-cell sequencing datasets. Spectral methods on a multilayer graph may then help cluster the cells by integrating information across individual graphs. One promising approach is the Spectral Clustering on Multi-Layer graphs (SCML) algorithm that finds a compromise among the low-frequency eigenspaces of individual graphs represented as points on a Grassmannian manifold (Dong *et al.*, 2013). The SCML algorithm seeks a consensus subspace that does not deviate too much from any one of the individual eigenspaces on this manifold.

The main goal of this paper is to unify the SCML (Dong *et al.*, 2013) and the WNN (Hao *et al.*, 2021) algorithms, two seemingly different approaches to identifying communities on multilayer graphs, into a common mathematical framework. We first propose two versions of Hamiltonian on a single-layer graph, related to spectral clustering and locally linear embedding (Roweis and Saul, 2000), respectively, and demonstrate that each Hamiltonian and an associated density matrix of mixed eigenstates provide a suitable representation of the graph. Intuitively, the density matrix encodes the correlation structure of nodes in the spectral embedding space obtained from the low-frequency eigenvectors of the Hamiltonian, while the Hamiltonian itself encodes the local connectivity of nodes on the original graph. Using this formalism, we interpret the SCML algorithm and reformulate the WNN analysis as optimization problems involving a set of Hamiltonian operators and density matrices from the individual layers of a multilayer graph, thereby providing a unified theoretical foundation for these methods and explaining their similarities and differences. Applying the SCML algorithm and the reformulated WNN method to a single-cell CITE-seq dataset consisting of human cord blood mononuclear cells (CBMCs) (Stoeckius *et al.*, 2017) yields results comparable to those obtained using the original WNN analysis.

2 Methods

2.1 Mathematical structure of graphs

A graph is a pair $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_i\}_{i=1}^N$ is the set of nodes (vertices), and $\mathcal{E}$ the set of links (edges) between nodes. A simple graph (without self-loops and multiple edges) can be described by an adjacency matrix $\mathbf{A}$, with the element $A_{ij} \geq 0$ representing the strength of the link from node v_i to node v_j; $A_{ij} = 0$ means the link does not exist. This paper considers undirected weighted graphs only and thus assumes $A_{ij} = A_{ji}$. The nodes often represent data points, and the weighted edges represent some measure of pairwise similarity between the data points.

The graph Laplacian is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where the degree matrix $\mathbf{D}$ is diagonal with entries

$$\mathbf{D}_{ii} = \mathrm{Deg}(v_i) \equiv \sum_{j=1}^{N} A_{ij} > 0, \;\; i = 1, \ldots, N.$$

The graph Laplacian is a discrete version of the Laplace operator on manifolds (Chung, 1997; Belkin and Niyogi, 2003; von Luxburg, 2007), up to a minus sign. The set of all real-valued functions on the vertex set $\mathcal{V}$ is isomorphic to $\mathbb{R}^N$. Thus, M functions $\mathbf{f}_m : \mathcal{V} \to \mathbb{R}$, $m = 1, \ldots, M$, defined on $\mathcal{V}$ can be organized into an $N \times M$ matrix $F = [\mathbf{f}_1, \ldots, \mathbf{f}_m, \ldots, \mathbf{f}_M]$, which may be viewed as a $\mathbb{R}^M$-valued function or a vector field on $\mathcal{V}$. Let $F(v_i)$ denote the i-th row of this matrix. Applying the graph Laplacian $\mathbf{L}$ to this matrix gives

$$(\mathbf{L}F)_{im} = \mathrm{Deg}(v_i)\, F_{im} - \sum_{j=1}^{N} A_{ij} F_{jm} = \sum_{j=1}^{N} A_{ij}(F_{im} - F_{jm}),$$

and since we assume that $\mathbf{A} = \mathbf{A}^t$, we have

$$\mathrm{tr}\left[F^{\mathrm{T}}\, \mathbf{L} F\right] = \frac{1}{2} \sum_{i,j=1}^{N} A_{ij} \|F(v_i) - F(v_j)\|_2^2\,, \tag{1}$$

which is just the sum of squared difference of vectors at two nodes weighted by the adjacency A_{ij}. It resembles the kinetic energy for a vector field and indicates how smoothly its components are varying on the graph.

A multilayer graph of $s_{\max}$ layers, $\mathcal{G} = \{(\mathcal{V}, \mathcal{E}^{(s)})\}_{s=1}^{s_{\max}}$, is a set of $s_{\max}$ graphs sharing a common node set $\mathcal{V}$. The individual graphs, $(\mathcal{V}, \mathcal{E}^{(s)})$, are referred to as graph layers and represent different connectivity between the nodes in $\mathcal{V}$, as described by the respective adjacency matrices $\{\mathbf{A}^{(s)}\}_{s=1}^{s_{\max}}$ or, equivalently, the graph Laplacian operators $\{\mathbf{L}^{(s)}\}_{s=1}^{s_{\max}}$. In the context of multi-omic single-cell datasets, different graph layers may encode the similarity of cells measured by different modalities. For example, from CITE-seq data, we can calculate two graph Laplacians from the RNA layer and the protein layer (Fig. 1A). The diagonal terms

come from the degree matrix, and different nodes, or even the same node in different layers, may have different degrees. The off-diagonal terms arising from subtracting the adjacency matrix are non-positive. Different layers of a multilayer graph may appear quite different, as links present in one graph may be absent in others (Fig. 1A).

One simple method of integrating the layers would be to take the arithmetic mean of the graph Laplacians $\{\mathbf{L}^{(s)}\}_{s=1}^{s_{\max}}$. However, this approach would amount to averaging the pairwise similarities without considering the higher-order structure of individual graphs. Other approaches include making the graph Laplacians commute approximately (Bronstein *et al.*, 2013), or equivalently, jointly diagonalizing the graph Laplacians approximately (Eynard *et al.*, 2015), as well as finding the geometric mean of the graph Laplacians (El Gheche *et al.*, 2020).

Motivated by the SCML algorithm (Dong *et al.*, 2013), this paper proposes a spectral method for merging multiple graph layers, each of which is represented by a Hamiltonian operator and a mixture of its eigenstates. The following section introduces two ways of constructing the Hamiltonian for a single-layer graph, and then justifies using the density matrix to capture the clustering structure of nodes on the graph.

2.2 Spectral analysis of graphs and mixture of eigenstates

As seen in Equation (1), the graph Laplacian behaves like a discrete version of the Laplace operator on functions defined on the set of nodes. Starting from the graph Laplacian, we construct two different Hamiltonian operators corresponding to the different optimization problems associated with the SCML and Seurat WNN algorithms.

The first Hamiltonian is closely related to spectral clustering and the problem of minimizing weighted cuts on graphs (von Luxburg, 2007; Meilă and Pentney, 2007; Dhillon *et al.*, 2007):

$$\mathbf{H}_{\mathrm{SC}} \equiv \mathbf{T}^{-1/2}\,\mathbf{L}\,\mathbf{T}^{-1/2}, \tag{2}$$

where $\mathbf{T} = \mathrm{diag}\{t_i\}_{i=1}^{N}$, $t_i > 0$, $\forall i$, is a diagonal matrix of positive values describing some notion of "importance" of the corresponding N nodes. Without further information about a graph, there are two common choices for $\mathbf{T}$: one choice is

$$\mathbf{T} = \mathbf{I} \quad \Rightarrow \quad \mathbf{H}_{\mathrm{SC}} = \mathbf{L} = \mathbf{D} - \mathbf{A},$$

and calculating a low-frequency eigenspace of this (unnormalized) graph Laplacian provides an approximate solution to minimizing the ratio cuts on the graph (Chan *et al.*, 1994); the other option is

$$\mathbf{T} = \mathbf{D}, \quad \Rightarrow \quad \mathbf{H}_{\mathrm{SC}} = \mathbf{L}_{\mathrm{sym}} \equiv \mathbf{D}^{-1/2}\,\mathbf{L}\,\mathbf{D}^{-1/2}, \tag{3}$$

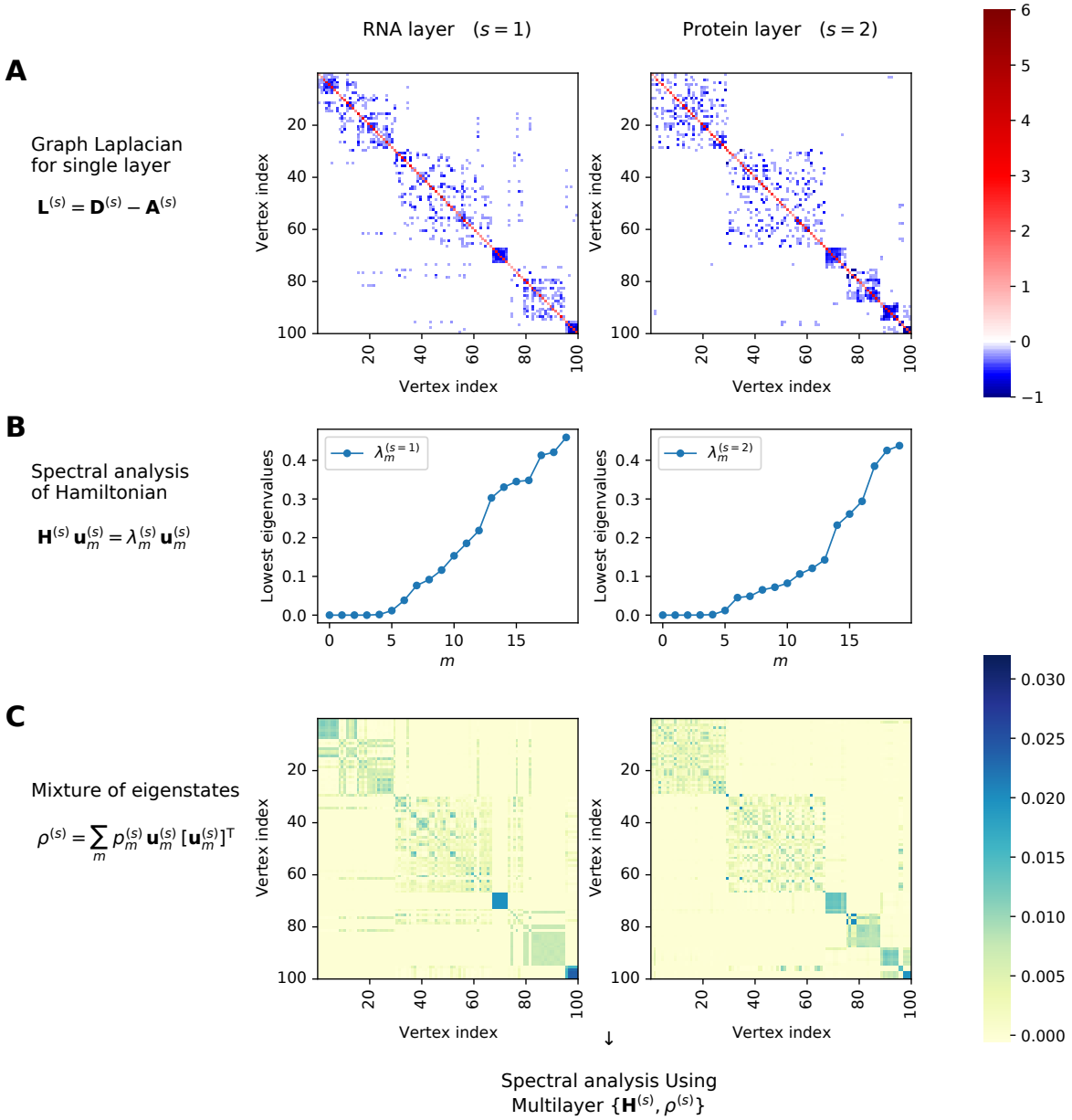

Figure 1: Illustration of the spectral analyses underlying our interpretation of the SCML method and reformulation of the WNN method. A subset of $N = 100$ cells were randomly selected from the CITE-seq data of human cord blood mononuclear cells. (A) Heatmap display of the graph Laplacian $\mathbf{L}^{(s)}$ constructed from the data in each modality. (B) The spectrum of the Hamiltonian $\mathbf{H}^{(s)}$ constructed from the graph Laplacian of each layer. The locally linear construction from Equation (4) with $\mathbf{W} = \mathbf{I}$ is shown. (C) Heatmap display of a density matrix representing a mixture of Hamiltonian eigenstates for each layer. The thermal mixture from Equation (8) is shown.

and calculating a low-frequency eigenspace of this symmetrically normalized graph Laplacian provides an approximate solution minimizing the normalized cuts on the graph (Shi and Malik, 2000; Stella and Shi, 2003). Generally, the eigenvectors corresponding to the ordered eigenvalues of $\mathbf{H}_{\mathrm{SC}}$, $0 \leq \lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{N-1}$, are

$$\mathbf{u}_0 = \underset{\mathbf{f}\in\mathbb{R}^N,\ \mathbf{f}^{\mathrm{T}}\mathbf{f}=1}{\arg\min} \frac{\mathbf{f}^{\mathrm{T}}\,\mathbf{L}\,\mathbf{f}}{\mathbf{f}^{\mathrm{T}}\,\mathbf{T}\,\mathbf{f}},$$

$$\mathbf{u}_m = \underset{\mathbf{f}\in\mathbb{R}^N,\ \mathbf{f}^{\mathrm{T}}\mathbf{f}=1, \mathbf{f}\perp\mathbf{u}_0,\ldots,\mathbf{u}_{m-1}}{\arg\min} \frac{\mathbf{f}^{\mathrm{T}}\,\mathbf{L}\,\mathbf{f}}{\mathbf{f}^{\mathrm{T}}\,\mathbf{T}\,\mathbf{f}}.$$

The second Hamiltonian motivated by the Locally Linear Embedding (LLE) algorithm (Roweis and Saul, 2000) is:

$$\mathbf{H}_{\mathrm{LL}} \equiv [\mathbf{W}\mathbf{L}_{\mathrm{rw}}]^{\mathrm{T}}\mathbf{W}\mathbf{L}_{\mathrm{rw}} = \mathbf{L}^{\mathbf{T}}\mathbf{D}^{-1}\mathbf{W}^2\mathbf{D}^{-1}\mathbf{L}, \tag{4}$$

where $\mathbf{L}_{\mathrm{rw}} \equiv \mathbf{D}^{-1}\mathbf{L}$ is the random walk graph Laplacian (Supplementary Methods), and $\mathbf{W} = \mathrm{diag}\{w_i\}_{i=1}^N$, $w_i \in [0,1]$, $\forall i$, is another diagonal matrix of node-specific weights. The random walk graph Laplacian computes the difference between a function's true value at a node and a weighted average of its values at immediate neighbors, as follows

$$(\mathbf{L}_{\mathrm{rw}}F)_{im} = F_{im} - \sum_j \frac{A_{ij}}{\mathrm{Deg}(v_i)}\ F_{jm} \equiv [(\mathbf{L}_{\mathrm{rw}}F)(v_i)]_m. \tag{5}$$

The Locally Linear Hamiltonian thus yields the quadratic form

$$\mathrm{tr}\left[F^{\mathrm{T}}\,\mathbf{H}_{\mathrm{LL}}\,F\right] = \sum_{i=1}^N w_i^2\ \|[\mathbf{L}_{\mathrm{rw}}F](v_i)\|_2^2\ ,$$

which sums the squared neighborhood prediction error at each node v_i weighted by w_i^2. Similar to $\mathbf{T}$, in the absence of further information, two natural choices for $\mathbf{W}$ are $\mathbf{W} = \mathbf{I}$ and $\mathbf{W} = \mathbf{D}$, yielding $\mathbf{H}_{\mathrm{LL}} = [\mathbf{L}_{\mathrm{rw}}]^{\mathrm{T}}\mathbf{L}_{\mathrm{rw}}$ and $\mathbf{H}_{\mathrm{LL}} = \mathbf{L}^2$, respectively. Learning these node-specific weights from data plays a key role in understanding the WNN analysis.

For either choice of the Hamiltonian operator $\mathbf{H}$, suppose the eigenvalues are ordered as $0 \leq \lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{N-1}$, with corresponding orthonormal eigenvectors $\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{N-1}$. The matrix

$$U = [\mathbf{u}_0, \ldots, \mathbf{u}_{K-1}], \tag{6}$$

forms the solution to the trace minimization problem

$$\min_{F\in\mathbb{R}^{N\times K}} \mathrm{tr}\left[F^{\mathrm{T}}\,\mathbf{H}\,F\right], \quad \text{s.t.} \quad F^{\mathrm{T}}F = \mathbf{I}. \tag{7}$$

For the Spectral Clustering Hamiltonian $\mathbf{H}_{\mathrm{SC}}$, the rows of U provide the spectral embedding (Belkin and Niyogi, 2003) and facilitate the spectral clustering (Ng *et al.*, 2001) of the nodes.

For the Locally Linear Hamiltonian $\mathbf{H}_{\mathrm{LL}}$, the rows of U offer the neighborhood-preserving mapping minimizing the reconstruction errors of the embedded coordinates (Roweis and Saul, 2000). The spectrum of $\mathbf{H}_{\mathrm{LL}}$, with $\mathbf{W} = \mathbf{I}$, for the CITE-seq data is shown in Fig. 1B.

We can represent a mixture of Hamiltonian eigenstates using a density matrix (a Hermitian positive semi-definite matrix with unit trace) as

$$\rho = \sum_{m=0}^{N-1} p_m \mathbf{u}_m \mathbf{u}_m^{\mathrm{T}}, \quad \text{where } p_m \geq 0, \forall m, \text{ and } \sum_{m=0}^{N-1} p_m = 1.$$

In particular, the density matrix $\rho = UU^{\mathrm{T}}/K$, recapitulating the low-frequency eigenspace representation in Equation (6), is the solution to the following trace minimization problem resembling Equation (7):

$$\min_{\rho} \operatorname{tr}[\rho \mathbf{H}], \quad \text{s.t.} \quad \|\rho\|_2 = \frac{1}{K},$$

where $\|\rho\|_2 = \max_m\{p_m\}$ is the spectral norm of ρ, and $K \in \mathbb{Z}^+$ is fixed. That is, a mixed state that minimizes the expected energy under the spectral norm constraint has a uniform mixture of the lowest K eigenstates (Supplementary Methods).

Alternatively, we can choose a probability distribution arising from the thermal equilibrium distribution in statistical physics, or the softmax function in machine learning:

$$p_m = e^{-\beta\lambda_m} \Big/ \sum_{m'=0}^{N-1} e^{-\beta\lambda_{m'}} \Rightarrow \quad \rho = \frac{e^{-\beta\mathbf{H}}}{\operatorname{tr}[e^{-\beta\mathbf{H}}]}, \tag{8}$$

where $\beta = 1/T$ is the inverse temperature. The thermal equilibrium density matrix is the solution to a trace minimization problem over density matrices with a different constraint,

$$\min_{\rho} \operatorname{tr}[\rho \mathbf{H}], \quad \text{s.t.} \quad -\operatorname{tr}[\rho \log \rho] = S,$$

where $S > 0$ is fixed entropy, and the Lagrange multiplier β^{-1} satisfies the relation $S = \beta \operatorname{tr}[\rho \mathbf{H}] + \log \operatorname{tr}\left[e^{-\beta\mathbf{H}}\right]$ (Supplementary Methods). Equivalently, this distribution maximizes entropy given fixed ensemble average of energy and provides the most unbiased probabilistic description of biological data (Finnegan and Song, 2017).

For the RNA and protein layers of the CITE-seq data, Fig. 1C shows their thermal equilibrium density matrices using the respective $\mathbf{H}_{\mathrm{LL}}$. In the Results sections 3.1 and 3.2, we will make specific choices of $\{\mathbf{H}^{(s)}, \rho^{(s)}\}$ for each layer to interpret the SCML algorithm and reformulate the WNN analysis in a common mathematical framework.

To obtain a feature map from the density matrix $\rho = \sum_{m=0}^{N-1} p_m \mathbf{u}_m \mathbf{u}_m^{\mathrm{T}}$, we define

$$\Phi : \mathcal{V} \to \mathbb{R}^N, v_i \mapsto (\sqrt{p_0}\mathbf{u}_0(v_i), \ldots, \sqrt{p_{N-1}}\mathbf{u}_{N-1}(v_i)).$$

Using this notation, the density matrix can be expressed as $\rho = \Phi\Phi^{\mathrm{T}}$, encoding the correlation structure of nodes in the embedding space.

2.3 CBMC CITE-seq data preprocessing

The scRNA-seq unique molecular identifier (UMI) count matrix was obtained from the Gene Expression Omnibus (GEO): GSE100866, `GSE100866_CBMC_8K_13AB_10X-RNA_umi.csv.gz`. Human genes were identified as those that began with "HUMAN_", and human cells were identified as those with greater than 90% of UMI tags aligned to human genes, resulting in a filtered UMI matrix, $X \in \mathbb{Z}_{\geq 0}^{G \times N}$, of $G = 20400$ human genes and $N = 8005$ human cells. We normalized the filtered count matrix by library size to remove the effect of variable transcript counts, and applied a square root transform to reduce the effect of extreme values, as follows:

$$(X_{\text{norm}})_{ji} = \sqrt{\frac{X_{ji}}{\sum_{k=1}^{G} X_{ki}} * \text{median}_l \left(\sum_{k=1}^{G} X_{kl} \right)}.$$

Finally, we performed principal component analysis (PCA) to denoise the high dimensional normalized expression data by projecting each cell onto the top 30 principal components. PCA was performed on the transpose of X_{norm} using sklearn.decomposition.PCA with options (n_components = 30, random_state = 12345678).

The antigen-derive tag (ADT) UMI matrix was also obtained from the GEO, `GSE100866_CBMC_8K_13AB_10X-ADT_umi.csv.gz`. We filtered this matrix to contain only the human cells identified from the scRNA-seq UMI count matrix, resulting in a count matrix $X^{\text{ADT}} \in \mathbb{Z}_{\geq 0}^{P \times N}$, where $P = 13$ is the number of antibodies profiled and $N = 8005$ is the number of human cells. The cell ADT profiles were normalized using a centered log ratio (CLR) transform with a pseudo-count of 1 (Stoeckius *et al.*, 2017):

$$(X_{\text{CLR}}^{\text{ADT}})_{ji} = \ln \frac{X_{ji}^{\text{ADT}} + 1}{g(X_{:i}^{\text{ADT}} + 1)}, \tag{9}$$

where $g(X_{:i} + 1) \equiv \left(\prod_{k=1}^{P} (X_{ki}^{\text{ADT}} + 1) \right)^{1/P}$ is the geometric mean of pseudo count adjusted ADT UMIs for cell i.

2.4 Graph construction

We constructed a cell-cell similarity graph using a method motivated by (van Dijk *et al.*, 2018). First, a k nearest neighbor (k-NN) distance graph was constructed for each modality using the Euclidean distance between the cell normalized profiles (PCA for RNA and CLR transform for ADT). The nearest neighbor distance graph was obtained using sklearn.neighbors.NearestNeigherbors with options (n_neighbors = k, metric = "minkowski", p = 2). A similarity graph was obtained from the k-NN distance graph using an adaptive Gaussian radial basis function. To construct the similarity graph, let D denote the cell-cell pairwise distance matrix and $k\text{NN}_i(j)$ denote the jth neighbor of cell i for a given modality

(RNA or ADT). The nonzero entries of the corresponding similarity graph B were obtained as

$$B_{i,k\mathrm{NN}_i(j)} = \exp\left[-\left(\frac{D_{i,k\mathrm{NN}_i(j)}}{D_{i,k\mathrm{NN}_i(a)}}\right)^2\right]$$

where $a \in [1, N]$ is a hyperparameter that specified the adaptive kernel width. For all datasets, we used the values $k = 30$ and $a = 10$, as recommended by (van Dijk *et al.*, 2018). All other entries in the matrix were set to zero. A symmetric graph adjacency matrix $\mathbf{A}$ was obtained by symmetrizing the k-NN similarity matrix as $\mathbf{A} = \frac{1}{2}\left(\mathbf{B} + \mathbf{B}^{\mathrm{T}}\right)$.

2.5 Spectral clustering implementation

Spectral decomposition and clustering of a cell similarity graph was performed on the corresponding symmetric graph Laplacian defined in in Equation (3). The spectral decomposition of the symmetric graph Laplacian was performed using scipy.sparse.linalg.eigsh. For clustering into K clusters, K orthonormal eigenvectors with the smallest eigenvalues were stacked as columns in $U \in \mathbb{R}^{N\times K}$. The spectral embedding vectors for each cell were ℓ_2 normalized for the purpose of clustering (Ng *et al.*, 2001; von Luxburg, 2007)

$$(U_{\mathrm{norm}})_{ij} = \frac{U_{ij}}{\sqrt{\sum_{\ell=1}^{K} U_{i\ell}^2}}.$$

Clustering was performed using sklearn.cluster.KMeans with options (n_cluster = K, max_iter = 1000, n_init = 100, random_state = 12345678).

2.6 Spectral clustering on multilayer graphs (SCML)

The SCML method was implemented for CITE-seq data by first constructing the symmetric graph Laplacians for the RNA and ADT profiles ($\mathbf{L}_{\mathrm{sym}}^{\mathrm{RNA}}, \mathbf{L}_{\mathrm{sym}}^{\mathrm{ADT}}$). The K eigenvectors with smallest corresponding eigenvalues were then obtained for each of the symmetric graph Laplacians, yielding two unnormalized spectral embedding matrices ($U^{\mathrm{RNA}}, U^{\mathrm{ADT}}$). The modified SCML graph Laplacian (Dong *et al.*, 2013) with hyperparameter $\alpha \in \mathbb{R}^+$ was then constructed:

$$\mathbf{L}_{\mathrm{SCML}} = \mathbf{L}_{\mathrm{sym}}^{\mathrm{RNA}} + \mathbf{L}_{\mathrm{sym}}^{\mathrm{ADT}} - \alpha\left(U^{\mathrm{RNA}}\left[U^{\mathrm{RNA}}\right]^{\mathrm{T}} + U^{\mathrm{ADT}}\left[U^{\mathrm{ADT}}\right]^{\mathrm{T}}\right). \tag{10}$$

The spectral decomposition of the modified Laplacian was performed using Pytorch (torch.linalg.eigh) after first converting the modified Laplacian to a Pytorch Tensor object. Cell clustering was performed with the K eigenvectors obtained from this decomposition using the same normalization and clustering described above.

2.7 UMAP visualization

UMAP (McInnes *et al.*, 2018b) was used to visualize a 2-dimensional embedding of the ℓ_2-normalized SCML embedding vectors (Methods 2.5). The 2-dimensional UMAP projections were constructed using the umap-learn UMAP function (McInnes *et al.*, 2018a) with options (random_state = 12345678, n_neighbors = 30, set_op_mix_ratio = 1, local_connectivity = 1, min_dist = 0.3, metric = "cosine").

2.8 Silhouette scores

The full pairwise distance matrices for the RNA PCA and ADT CLR profiles were calculated using scipy.spatial.distance.pdist with options (metric = "euclidean") followed by scipy.spatial.distance.squareform to format the distances as a matrix. The silhouette scores for the RNA and ADT distance matrices were then obtained with sklearn.metrics.silhouette_score with option (metric = "precomputed").

2.9 Seurat analysis of the CBMC dataset

Using the same criteria as in Methods 2.3, we selected the cells with more than 90% RNA reads aligned to the human genome, and obtained the RNA and ADT count data for a total of $N = 8005$ human cells, removing the RNA count from the mouse genes (using the "CollapseSpeciesExpressionMatrix" function with ncontrols = 0). We also removed three insignificant ADTs (CCR5, CCR7, CD10), as previously suggested (Stuart *et al.*, 2019).

We then followed the standard Seurat workflow for CITE-seq data. Analysis of the RNA data involved: normalizing the count data (NormalizeData) using with the default normalization.method = "LogNormalize", scale.factor = 10000; finding the 2000 most variable features; scaling those selected features; and, running PCA. Analysis of the ADT data involved: normalizing the count data (NormalizeData) using normalization.method = "CLR", margin = 2; scaling all features; and, running PCA.

Lastly, we applied the WNN method, which first constructed nearest neighbor graphs on the first 30 PCs of the RNA data and the first 7 PCs of the ADT data, then computed the cell-specific modularity weights for each modality, and built a WNN graph, all incorporated in the "FindMultiModalNeighbors" function of Seurat v4 (Hao *et al.*, 2021). The WNN graph was used for dimension reduction and clustering: a UMAP embedding in two dimension, and a modularity-based clustering with resolution 0.2, rendering $K = 12$ clusters. The clusters are then manually labeled based on their marker genes and proteins (using the "FindAllMarkers" function).

3 Results

3.1 SCML penalizes the Hilbert-Schmidt distance between density matrices

In the SCML algorithm for a multilayered graph with N vertices (Dong *et al.*, 2013), the K-dimensional lowest-frequency eigenspace $U \in \mathbb{R}^{N\times K}$ of each normalized graph Laplacian is represented as a point on the Grassmann manifold $\mathrm{Gr}_K(\mathbb{R}^N)$ of all K-dimensional subspaces in $\mathbb{R}^N$. The squared chordal distance between two subspaces U and U' on the manifold is

$$d^2(U,U') \equiv \frac{1}{2}\left\|UU^{\mathrm{T}} - U'U'^{\mathrm{T}}\right\|_F^2 = K - \mathrm{tr}\left[UU^{\mathrm{T}}\,U'U'^{\mathrm{T}}\right].$$

Given the individual eigenspaces $U^{(s)}$ of $s_{\max}$ layers, the SCML algorithm searches for a consensus subspace U that minimizes the joint spectral energy (Equation (7)) of all layers while also minimizing the sum of squared distance to the eigenspace $U^{(s)}$ of each layer, by adding the term

$$\sum_{s=1}^{s_{\max}} d^2(U,U^{(s)}) = \; s_{\max}\,K - \mathrm{tr}\left[UU^{\mathrm{T}}\;\sum_{s=1}^{s_{\max}} U^{(s)}[U^{(s)}]^{\mathrm{T}}\right]$$

to the loss function, resulting in a new trace minimization problem with the Hamiltonian

$$\mathbf{H}_{\mathrm{SCML}} = \sum_{s=1}^{s_{\max}} \mathbf{H}_{\mathrm{SC}}^{(s)} - \alpha \sum_{s=1}^{s_{\max}} U^{(s)}[U^{(s)}]^{\mathrm{T}}, \tag{11}$$

where $\mathbf{H}_{\mathrm{SC}}^{(s)}$ is the Spectral Clustering Hamiltonian (normalized graph Laplacian) of each layer (Equation (2)), and $\alpha \in \mathbb{R}^+$ is a hyperparameter balancing the two terms in the minimization

$$\min_{U\in\mathbb{R}^{N\times K}} \mathrm{tr}\left[U^{\mathrm{T}}\,\mathbf{H}_{\mathrm{SCML}}\,U\right], \quad \text{s.t.} \quad U^{\mathrm{T}}U = \mathbf{I}.$$

For CITE-seq data, we have $s_{\max} = 2$ ($s = 1$ for RNA, $s = 2$ for ADT), and the SCML Hamiltonian $\mathbf{H}_{\mathrm{SCML}}$ reproduces Equation (10) when the symmetrically normalized graph Laplacian is chosen for each $\mathbf{H}_{\mathrm{SC}}^{(s)}$.

Note that for two density matrices $\rho = \frac{1}{K}UU^{\mathrm{T}}$, $\sigma = \frac{1}{K}U'U'^{\mathrm{T}}$ that are uniform mixtures of K orthonormal states, their Hilbert-Schmidt distance is equivalent to the squared distance $d^2(U,U')$ on the Grassmannian manifold:

$$D_{\mathrm{HS}}(\rho,\sigma) \equiv \mathrm{tr}\left[(\rho-\sigma)^2\right] = \frac{2}{K} - 2\,\mathrm{tr}\left[\rho\sigma\right] = \frac{2}{K^2}\,d^2(U,U').$$

Using the uniform-mixture density matrices $\rho^{(s)}$ associated with the K-dimensional lowest-frequency eigenspace of each $\mathbf{H}_{\mathrm{SC}}^{(s)}$, the SCML Hamiltonian becomes

$$\mathbf{H}_{\mathrm{SCML}} = \sum_{s=1}^{s_{\max}} \mathbf{H}_{\mathrm{SC}}^{(s)} - \alpha K \sum_{s=1}^{s_{\max}} \rho^{(s)}, \tag{12}$$

and the SCML algorithm amounts to finding a density matrix ρ that solves

$$\min_{\rho} \operatorname{tr}\left[\rho\, \mathbf{H}_{\mathrm{SCML}}\right] = \min_{\rho} \sum_{s=1}^{s_{\max}} \left\{ \operatorname{tr}\left[\rho\, \mathbf{H}_{\mathrm{SC}}\right] + \frac{K}{2} \alpha D_{\mathrm{HS}}(\rho, \rho^{(s)}) \right\},$$

where ρ is again constrained to satisfy $\|\rho\|_2 = 1/K$. As in $\mathbf{H}_{\mathrm{SC}} = \mathbf{T}^{-1}(\mathbf{D} - \mathbf{A})\mathbf{T}^{-1}$ where the off-diagonal contains the normalized pairwise adjacency, the magnitude of the off-diagonal terms in $\mathbf{H}_{\mathrm{SCML}}$ (Equation (12)) also represents the modified affinity between nodes,

$$a_{ij} = \sum_{s=1}^{s_{\max}} a_{ij}^{(s)} + \alpha K \sum_{s=1}^{s_{\max}} \rho_{ij}^{(s)}, \quad \text{where } a_{ij}^{(s)} \equiv \frac{A_{ij}^{(s)}}{\sqrt{t_i^{(s)} t_j^{(s)}}}. \tag{13}$$

We can thus interpret the SCML algorithm as computing the modified affinity a_{ij} between nodes on a multilayer graph, using the affinity $a_{ij}^{(s)}$ as well as the correlation $\rho_{ij}^{(s)}$ of spectral embedding in each layer s.

In next section, we will use the same language of Hamiltonian operators and density matrices to reformulate the WNN method as computing a different form of affinity between nodes on a multilayer graph.

3.2 Multilayer spectral graph theory provides a mathematical foundation for the WNN method

The WNN analysis aims to compute the weighted affinity between graph nodes based on their affinity in each modality (Hao *et al.*, 2021), as follows:

(1) For each modality, compute a similarity graph with an adjacency matrix $\mathbf{A}^{(s)}$ and a low-frequency representation $\Phi^{(s)}$ of the nodes.

(2) Calculate the nearest neighbor prediction error for layers s_1 and s_2:

$$\Phi^{(s_1)}(v_i) - \frac{\sum_j A_{ij}^{(s_2)}\, \Phi^{(s_1)}(v_j)}{\sum_j A_{ij}^{(s_2)}}. \tag{14}$$

If $s_1 = s_2$, it is the within-modality nearest neighbor prediction error; otherwise, it is the cross-modality prediction error. A key observation used in our work is that we can rewrite the above prediction error as $[\mathbf{L}_{\mathrm{rw}}^{(s_2)}\, \Phi^{(s_1)}](v_i)$, as in Equation (5).

(3) From the within- and cross-modality nearest neighbor prediction errors, learn a set of modality weights $\{w_i^{(s)}\}_{s=1}^{s_{\max}}$ for each node v_i, satisfying

$$\sum_{s=1}^{s_{\max}} w_i^{(s)} = 1, \quad \forall i. \tag{15}$$

(4) Calculate the weighted affinity from v_i to v_j, using the learned node-specific modality weights for v_i,

$$a_{ij} = \sum_{s=1}^{s_{\max}} w_i^{(s)} a_{ij}^{(s)}, \tag{16}$$

where $a_{ij}^{(s)}$ is some notion of affinity from v_i to v_j on graph layer s. Finally, a WNN graph is constructed from the weighted affinity matrix and used for downstream analysis, such as dimension reduction and clustering.

We see that in Step (2) of the WNN analysis, the random walk graph Laplacian naturally arises, suggesting that multilayer spectral graph theory may provide a rigorous mathematical foundation for the WNN method. In our ensuing interpretation of the algorithm, we make choices that utilize spectral techniques and deviate slightly from the Seurat v4 implementation (Hao *et al.*, 2021). For example, the original method uses the eigenvectors from the principal component analysis (PCA) of each modality data as the representation $\Phi^{(s)}$ of the layer ; here, we instead use a mixture of the eigenstates of a Hamiltonian operator constructed from the random walk graph Laplacian (Methods). We also formulate an optimization problem to solve for the node-specific modality weights. Despite these differences, our implementation below provides results very similar to those from the original WNN analysis.

For each modality, we first construct a similarity graph with an adjacency matrix $\mathbf{A}^{(s)}$ (Methods) and calculate the random walk graph Laplacian $\mathbf{L}_{\mathrm{rw}}^{(s)}$. Without any prior knowledge about the graphs, we first use the identity weight matrix in the Locally Linear Hamiltonian (Equation (4)) and the thermal mixture of its eigenstates to initialize the representation of each layer:

$$\mathbf{H}_0^{(s)} = [\mathbf{L}_{\mathrm{rw}}^{(s)}]^{\mathrm{T}}\, \mathbf{L}_{\mathrm{rw}}^{(s)}, \quad \rho^{(s)} = \frac{e^{-\beta \mathbf{H}_0^{(s)}}}{\mathrm{tr}\left[e^{-\beta \mathbf{H}_0^{(s)}}\right]} \equiv \Phi^{(s)} \Phi^{(s)\mathrm{T}}, \tag{17}$$

where the feature map $\Phi^{(s)}$ from the density matrix $\rho^{(s)}$ embeds the set of N nodes in layer s into $\mathbb{R}^N$ (Methods), similar to Step (1) above.

To learn the node-specific modality weights, we next modify the Locally Linear Hamiltonian to include a diagonal weight matrix, as

$$\mathbf{H}_{\mathrm{LL}}^{(s)} = \mathbf{L}_{\mathrm{rw}}^{(s)\mathrm{T}}\, [\mathbf{W}^{(s)}]^2\, \mathbf{L}_{\mathrm{rw}}^{(s)},$$

where $\mathbf{W}^{(s)} = \mathrm{diag}\{w_i^{(s)}\}_{i=1}^N$, $w_i^{(s)} \in [0,1]$, $\forall i, s$, with the diagonal elements satisfying Equation (15) as in Step (3) of WNN. Note that, by construction, $\mathbf{H}_{\mathrm{LL}}^{(s)}$ computes the weighted sum of squared within- or cross-modality prediction errors for the above feature maps (Equa-

tion (17)),

$$\begin{aligned}
&\operatorname{tr}\left[\rho^{(s_1)}\,\mathbf{H}_{\mathrm{LL}}^{(s_2)}\right] = \operatorname{tr}\left\{\Phi^{(s_1)}\Phi^{(s_1)\mathrm{T}}\mathbf{L}_{\mathrm{rw}}^{(s)\mathrm{T}}\left[\mathbf{W}^{(s)}\right]^2\mathbf{L}_{\mathrm{rw}}^{(s)}\right\} \\
&= \sum_{i=1}^{N}\left[w_i^{(s_2)}\right]^2 \left\|\left[\mathbf{L}_{\mathrm{rw}}^{(s_2)}\,\Phi^{(s_1)}\right](v_i)\right\|_2^2 = \sum_{i=1}^{N}\left[w_i^{(s_2)}\,\Delta_i^{(s_1,s_2)}\right]^2,
\end{aligned}$$

where $\Delta_i^{(s_1,s_2)} \equiv \|[\mathbf{L}_{\mathrm{rw}}^{(s_2)}\,\Phi^{(s_1)}](v_i)\|_2$. The lower the value of $\Delta_i^{(s_1,s_2)}$, the better we can use the neighboring nodes of v_i in layer s_2 to predict its function value at v_i in layer s_1. Thus, the difference between cross- and within-modality prediction errors, weighted by $\left[w_i^{(s_2)}\right]^2$, is

$$\begin{aligned}
&\operatorname{tr}\left\{\left[\rho^{(s_1)} - \rho^{(s_2)}\right]\mathbf{H}_{\mathrm{LL}}^{(s_2)}\right\} \\
&= \sum_{i=1}^{N}\left[w_i^{(s_2)}\right]^2\left\{\left[\Delta_i^{(s_1,s_2)}\right]^2 - \left[\Delta_i^{(s_2,s_2)}\right]^2\right\};
\end{aligned}$$

and, adding a similar term after switching s_1 and s_2 yields

$$\begin{aligned}
&-\operatorname{tr}\left\{\left[\rho^{(s_1)} - \rho^{(s_2)}\right]\left[\mathbf{H}_{\mathrm{LL}}^{(s_1)} - \mathbf{H}_{\mathrm{LL}}^{(s_2)}\right]\right\} \\
&= \sum_{i=1}^{N}\left[w_i^{(s_1)}\right]^2\left\{\left[\Delta_i^{(s_2,s_1)}\right]^2 - \left[\Delta_i^{(s_1,s_1)}\right]^2\right\}, \\
&+ \sum_{i=1}^{N}\left[w_i^{(s_2)}\right]^2\left\{\left[\Delta_i^{(s_1,s_2)}\right]^2 - \left[\Delta_i^{(s_2,s_2)}\right]^2\right\}.
\end{aligned}$$

To implement Steps (2) and (3) of the WNN analysis, we propose to learn the node-specific modality weights by minimizing this difference. When $s_{\max} = 2$, we thus need to solve (Supplementary Methods)

$$\begin{aligned}
&\mathbf{W}_*^{(1)}, \mathbf{W}_*^{(2)} \\
&= \underset{\mathbf{W}^{(1)}+\mathbf{W}^{(2)}=\mathbf{I}}{\arg\max}\ \operatorname{tr}\left\{\left[\rho^{(1)} - \rho^{(2)}\right]\left[\mathbf{H}_{\mathrm{LL}}^{(1)} - \mathbf{H}_{\mathrm{LL}}^{(2)}\right]\right\}.
\end{aligned} \tag{18}$$

This scheme can be generalized to the case of more than two layers by iterating over each pair of layers

$$\underset{\sum_{s=1}^{s_{\max}}\mathbf{W}^{(s)}=\mathbf{I}}{\arg\max}\ \sum_{(s_1,s_2)}\operatorname{tr}\left\{\left[\rho^{(s_1)} - \rho^{(s_2)}\right]\left[\mathbf{H}_{\mathrm{LL}}^{(s_1)} - \mathbf{H}_{\mathrm{LL}}^{(s_2)}\right]\right\},$$

while keeping the sum of node-specific weights to be one.

Lastly, taking the affinity in Equation (16) to be $a_{ij}^{(s)} = A_{ij}^{(s)}/\sum_j A_{ij}^{(s)}$, the final WNN multilayer graph can be described by the following consensus random walk graph Laplacian,

$$\mathbf{L}_{\mathrm{rw}} = \sum_{s=1}^{s_{\max}}\mathbf{W}_*^{(s)}\mathbf{L}_{\mathrm{rw}}^{(s)},$$

and the final Weighted Locally Linear Hamiltonian for the combined similarity graph using this consensus $\mathbf{L}_{\mathrm{rw}}$ is

$$\mathbf{H}_{\mathrm{WLL}} = \mathbf{L}_{\mathrm{rw}}^{\mathrm{T}} \mathbf{L}_{\mathrm{rw}}, \tag{19}$$

which can then be used to solve for a K-dimensional feature map Φ for further downstream analysis (Methods).

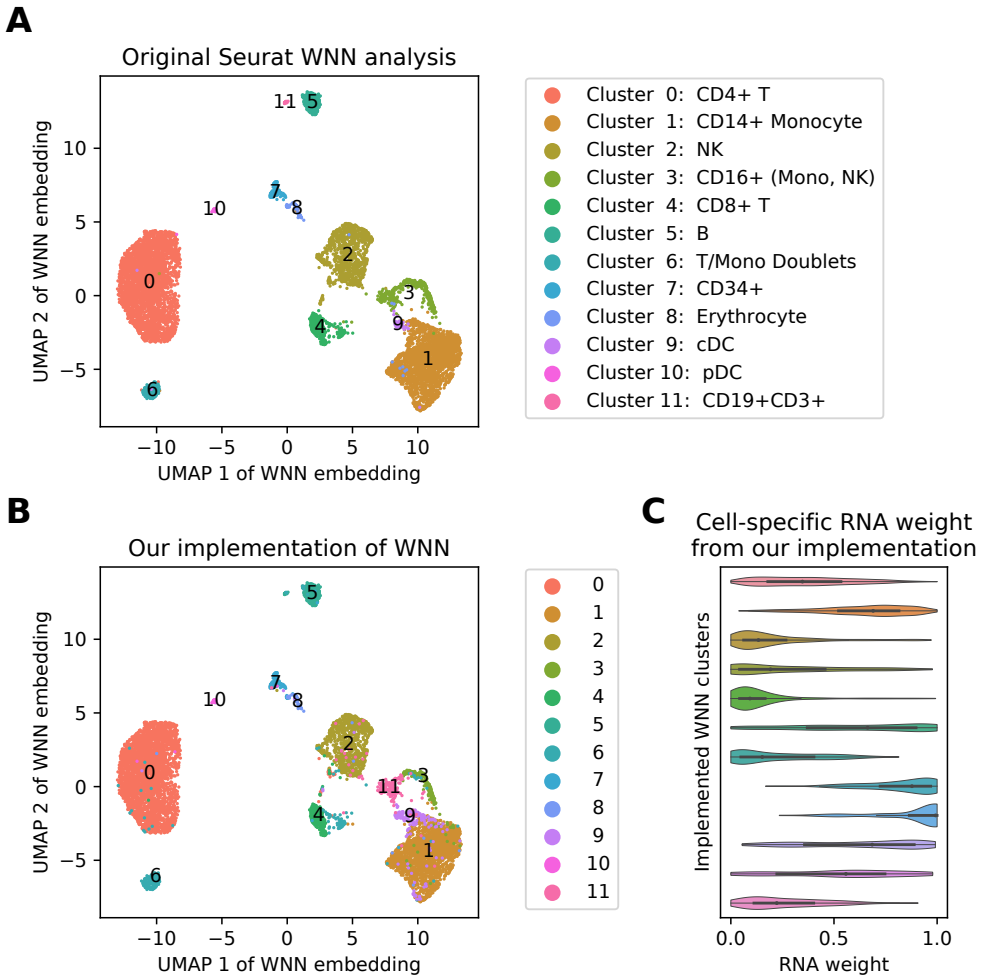

Figure 2: Comparison of our implementation of the WNN method with the original algorithm in Seurat v4. (A) UMAP view of the original Seurat WNN result, and a modularity-based clustering with resolution 0.2, resulting in $K = 12$ clusters, which are then manually labeled based on the marker genes and proteins. (B) The same UMAP view of the weighted locally linear (WLL) similarity graph generated by our implementation of the WNN method. Using a feature map corresponding to the $K = 12$ lowest Hamiltonian eigenstates, spectral clustering with K-means yields similar clustering as in (A). (C) Cell-specific modality weights obtained from our implementation show different distribution in different cell types identified in (B).

To compare our reformulation with the original Seurat WNN method, we applied both

approaches to the human CBMC CITE-seq dataset (Stoeckius *et al.*, 2017). This dataset included 5% spiked-in mouse cells, which we removed from the analysis to reduce the influence of highly distinct but biologically uninteresting cell types (Fig. 2A, Methods). The Seurat WNN clustering was performed using resolution of 0.2 (Methods), yielding 12 cell types consistent with the previously reported results (Hao *et al.*, 2021). For comparison, we fixed the number of clusters to be 12 for the reformulated WNN algorithm (Fig. 2B). While our implementation of the WNN idea deviated from the Seurat software package in technical details, the clustering results were nevertheless very similar. We also showed the cell-specific weights of the RNA modality in Fig. 2C. As in the original WNN results (Hao *et al.*, 2021), if the surface markers of a certain cell type were not measured in the ADT assay, then their RNA weights tended to be larger; conversely, the presence of specific surface marker measurements made the cells' RNA weights smaller than the ADT weights.

3.3 WNN and SCML methods yield comparable results

We compared the Seurat WNN method with graph spectral clustering approaches using the CBMC CITE-seq dataset (Stoeckius *et al.*, 2017) (Methods). We partitioned the cells into 12 clusters using each of RNA spectral, ADT spectral, and SCML clustering algorithms to match the number of clusters obtained with the WNN algorithm (Methods, Supplementary Methods, Supplementary Figures 1 and 2). The clustering results for each of these four algorithms were projected onto the two-dimensional UMAP obtained from the SCML embedding vectors (Methods, Fig. 3).

We calculated the silhouette scores for both RNA and ADT pairwise distance matrices to quantify the separation of the clusters obtained using each of the four methods (Methods, Table 1). The silhouette scores demonstrated that both SCML and Seurat WNN approaches achieved a compromise between the clustering results based on either RNA or ADT alone. That is, compared to the ADT-based spectral clustering that showed a poor silhouette score for the RNA distance matrix, both integrative approaches noticeably improved the score for the RNA distance, while still maintaining a high score for the ADT distance matrix (Table 1). Similarly, compared to the RNA-based spectral clustering, both integrative approaches improved the score for the ADT distance, although the score for the RNA distance matrix decreased slightly, because the CD4+ and CD8+ T cell clusters were correctly separated by the WNN and SCML methods, as further discussed below (Table 1, Supplementary Table 1).

This comparison suggested that both integrative approaches produced a clustering that compromised between the two modalities and thus likely yielded a better separation of true biological cell types, which might have been blurred by experimental biases and artifacts such as the scRNA-seq dropout effects and the limited number of profiled ADT surface

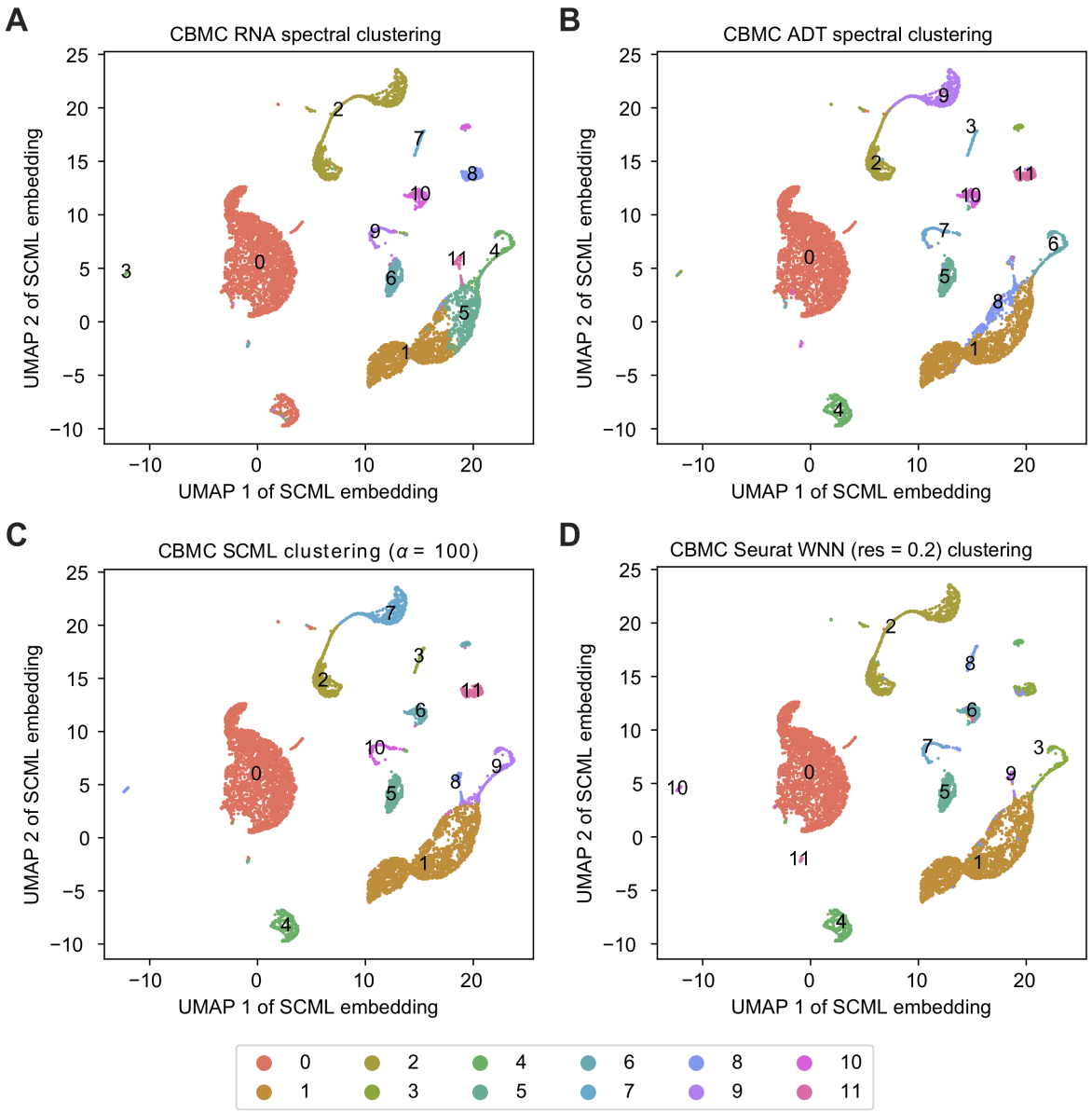

Figure 3: Comparison of spectral and weighted nearest neighbor clustering algorithms on the CBMC dataset. (**A**) Spectral clustering using only RNA data. (**B**) Spectral clustering using only ADT data. (**C**) SCML clustering algorithm with hyperparameter $\alpha = 100$. (**D**) Seurat WNN algorithm with clustering hyperparameter resolution = 0.2.

proteins. For example, the RNA-based spectral clustering grouped the clusters labeled by the WNN method as CD4+ helper and CD8+ cytotoxic T cells (Seurat WNN clusters 0 and 4, respectively) (Fig. 2, Fig. 3) which were clearly separated in the ADT space. By contrast, both Seurat WNN and SCML algorithms correctly partitioned these cell types. Merging the CD4+ and CD8+ T cell clusters together improved the RNA silhouette scores, but worsened the ADT silhouette scores for both integrative methods (Supplementary Table

1), demonstrating that the ideal clustering for one modality may not correspond to that for the other and that a compromise is needed to identify biologically distinct cell types.

Table 1: CBMC mean silhouette scores for different clustering methods and distance measures

Clustering method	RNA distance	ADT distance
RNA spectral	0.3696	0.2496
ADT spectral	0.08461	0.3941
SCML ($\alpha = 100$)	0.2177	0.4211
Seurat WNN (res = 0.2)	0.2406	0.3977

Comparing the clustering results obtained using the SCML algorithm (with $\alpha = 100$) to those obtained using Seurat's WNN algorithm (with resolution = 0.2) showed that some of the major differences involved the partitioning of natural killer (NK) cells (Seurat cluster 2) and CD16+ cells (Seurat cluster 3) (Fig. 2, Fig. 3). Each of these two cell types were further partitioned into two clusters by the SCML algorithm. The NK cells were also partitioned into two clusters by the ADT spectral clustering, but not the RNA spectral clustering, suggesting that the partitioning resulted primarily from differences in the ADT profile of cell subpopulations. Comparing ADT markers between the two partitions of the NK cells (SCML clusters 2 and 7) showed that cells in SCML cluster 7 exhibited higher levels of CD8 compared to cluster 2 (Supplementary Figure 3). A separation of the NK cells into CD8+ and CD8- subtypes in this dataset was previously described by Hao *et al.*, 2021, and we also observed this separation when using a higher resolution for the WNN clustering (Supplementary Figure 4), demonstrating that the sub-partitioning of the NK cells by the SCML algorithm was actually biologically meaningful. The CD16+ cluster (Seurat cluster 3) was also partitioned into two clusters, 9 and 11, by the SCML algorithm; SCML cluster 11 exhibited a larger ADT library size than SCML cluster 9 (median total ADT UMI counts of 18008 and 2735, respectively) and elevation of CD8 and CD45RA in the CLR transformed profiles. The CD16+ cells were also partitioned into two clusters by both RNA-based and ADT-based spectral clustering, suggesting that these differences were present in both transcriptional and protein-level profiles. As was the case for the NK cells, the Seurat WNN algorithm also partitioned the CD16+ cells into two clusters, similar to those obtained by the SCML algorithm, when a larger resolution value was used (Supplementary Figure 4).

4 Discussion and Conclusion

In this paper, we have reformulated two algorithms, SCML and Seurat WNN, for clustering single-cell multi-omic datasets in a unified framework of multilayer spectral graph analysis. We have shown that these algorithms can be expressed as optimization problems involving different Hermitian operators acting on functions defined on the graph nodes. Our results highlight the connection between these seemingly distinct algorithms and provide a mathematical framework for understanding these multi-modal clustering algorithms.

Applying the SCML, Seurat WNN, and spectral clustering based on RNA or ADT alone to a single-cell CITE-seq dataset of cord blood mononuclear cells (Stoeckius *et al.*, 2017) has shown that the SCML and Seurat WNN algorithms yield comparable clustering results, recapitulating the biologically meaningful separation of cell types and, at the same time, offering reconciliation of potential differences between the RNA and ADT modalities (Fig. 3, Table 1). Apparent differences between SCML and WNN results may inevitably arise because of the differences in their optimization functions and algorithmic details, but some of these differences may get resolved when resolution parameters and cluster numbers are adjusted. This result suggests that an ideal clustering of cells based on one modality may not always agree with that based on another modality and that an integrative clustering approach incorporating all available information better partitions the cells into biological cell types. Our result thus highlights the benefit of multi-omics research, as a single modality may be prone to noise or provide an incomplete description of single cells.

A limitation of the SCML algorithm is that it relies on the assumption that the majority of the modalities are informative of the true underlying clusters (Dong *et al.*, 2013). If one of the modalities is very noisy, including this modality in the joint clustering may actually make it difficult to discover the true clusters detected by other modalities. The graph and embedding subspace obtained from the noisy modality may not capture the true cluster structure, but nevertheless will pull the consensus subspace towards the incorrect subspace on the Grassmannian manifold. Furthermore, if the clusters obtained from different modalities represent radically different partitions of the datasets, then finding a subspace between the individual embedding subspaces on the Grassmannian manifold may not represent a useful clustering of the dataset. This limitation is not, however, unique to the SCML algorithm (Hao *et al.*, 2021) and represents an assumption fundamental to many methods trying to integrate information across multiple modalities. By using a local modality weight, the Seurat WNN algorithm (Hao *et al.*, 2021) may be able to handle differences in the informativeness of each modality for different cell types better than the SCML algorithm; however, while it may certainly be that cell-specific modality weights might aid in clustering certain cell types (Hao *et al.*, 2021), we have shown that the SCML algorithm can uncover many of the same clusters without the need of cell-specific weights.

Funding

This project was supported in part by grants from the National Institutes of Health (R01CA163336, R01HD089552).
Conflict of Interest: none declared.

References

Belkin, M. and Niyogi, P. (2003). Laplacian Eigenmaps for dimensionality reduction and data representation. *Neural Computation*, **15**(6), 1373–1396.

Bronstein, M. M. *et al.* (2013). Making Laplacians commute. *arXiv preprint arXiv:1307.6549*.

Canzar, S. *et al.* (2021). A generalization of t-SNE and UMAP to single-cell multimodal omics. *Genome Biology*, **22**(1), 130.

Chan, P. K. *et al.* (1994). Spectral k-way ratio-cut partitioning and clustering. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **13**(9), 1088–1096.

Chung, F. R. (1997). *Spectral graph theory*. Number 92. American Mathematical Society, Providence, Rhode Island.

Dhillon, I. S. *et al.* (2007). Weighted graph cuts without eigenvectors: a multilevel approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **29**(11), 1944–1957.

Dong, X. *et al.* (2013). Clustering on multi-layer graphs via subspace analysis on Grassmann manifolds. *IEEE Transactions on Signal Processing*, **62**(4), 905–918.

El Gheche, M. *et al.* (2020). OrthoNet: multilayer network data clustering. *IEEE Transactions on Signal and Information Processing over Networks*, **6**, 152–162.

Eynard, D. *et al.* (2015). Multimodal manifold analysis by simultaneous diagonalization of Laplacians. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **37**(12), 2505–2517.

Finnegan, A. and Song, J. S. (2017). Maximum entropy methods for extracting the learned features of deep neural networks. *PLoS Computational Biology*, **13**(10), e1005836.

Hao, Y. *et al.* (2021). Integrated analysis of multimodal single-cell data. *Cell*, **184**(13), 3573–3587.

Kim, H. J. *et al.* (2020). CiteFuse enables multi-modal analysis of CITE-seq data. *Bioinformatics*, **36**(14), 4137–4143.

Liu, Y. *et al.* (2020). Single-cell profiling reveals divergent, globally patterned immune responses in murine skin inflammation. *iScience*, **23**(10), 101582.

McInnes, L. *et al.* (2018a). UMAP: Uniform manifold approximation and projection. *Journal of Open Source Software*, **3**(29), 861.

McInnes, L. *et al.* (2018b). UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

Meilă, M. and Pentney, W. (2007). Clustering by weighted cuts in directed graphs. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, pages 135–144. SIAM.

Ng, A. Y. *et al.* (2001). On spectral clustering: Analysis and an algorithm. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, pages 849–856, Cambridge, MA, USA. MIT Press.

Peterson, V. M. *et al.* (2017). Multiplexed quantification of proteins and transcripts in single cells. *Nature Biotechnology*, **35**(10), 936–939.

Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by Locally Linear Embedding. *Science*, **290**(5500), 2323–2326.

Shahi, P. *et al.* (2017). Abseq: Ultrahigh-throughput single cell protein profiling with droplet microfluidic barcoding. *Scientific Reports*, **7**(1), 44447.

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**(8), 888–905.

Stella, X. Y. and Shi, J. (2003). Multiclass spectral clustering. In *Proceedings Ninth IEEE International Conference on Computer Vision*, volume 1, pages 313–319. IEEE Computer Society.

Stoeckius, M. *et al.* (2017). Simultaneous epitope and transcriptome measurement in single cells. *Nature Methods*, **14**(9), 865–868.

Stuart, T. *et al.* (2019). Comprehensive integration of single-cell data. *Cell*, **177**(7), 1888–1902.

van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, **9**, 2579–2605.

van Dijk, D. *et al.* (2018). Recovering gene interactions from single-cell data using data diffusion. *Cell*, **174**(3), 716–729.

von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, **17**(4), 395–416.

Wang, B. *et al.* (2014). Similarity network fusion for aggregating data types on a genomic scale. *Nature Methods*, **11**(3), 333–337.

Zhao, C. and Song, J. S. (2018). Quantum transport senses community structure in networks. *Physical Review E*, **98**(2), 022301.