Manifold Learning in Differentiating Cancer Cells

Can Chen * Yuting Liu† Peter Paquet ‡ James Connolly §

Abstract

In many complex systems, such as those arising in biology, capturing the interplay of function, structure, and dynamics is critical in order to characterize the underlying mechanism. Those representations are in large scale and high-dimensional. In order to understand the data better, dimensionality reduction techniques are necessary to employ. In this work, we analytically review two important nonlinear dimensionality reduction methods – Laplacian Eigenmaps and t-Distributed Stochastic Neighbor Embedding from the literature. At the end, we apply both methods to real-world genomic data from a University of Michigan breast cancer research laboratory.

1 Introduction: Motivation and Literature Review

In many complex biological and engineering systems, such as the human genome, data can be acquired at an unprecedented resolution by advanced technologies. These high-dimensional data are not easily interpreted as they cannot be viewed in two- or three-dimensional plots. Principle component analysis (PCA) is a popular tool that is used to reduce variables' dimensions while retaining key information from the high-dimensional data. However, PCA heavily relies on the assumption that the data are linearly correlated. We seek to perform data analysis, like feature extraction, over a dataset that is not linearly correlated (e.g. in spiral shape). To do this, manifold learning, also known as nonlinear dimensionality reduction, would greatly aid the case. Manifold learning treats the observed data as a manifold that is approximately isomorphic to a lower dimensional Euclidean space (1). In this project, we propose to investigate two powerful manifold learning methods – Laplacian Eigenmaps, introduced by Belkin (2) in 2002, and t-Distributed Stochastic Neighbor Embedding (t-SNE), introduced by Maaten (3) in 2008.

Laplacian Eigenmaps seeks to exploit the (assumed) geometric structure of the dataset by constructing a graph of the data and then employing spectral techniques on the graph to reduce the dataset's dimensionality (2). While the technique has appealing locality preservation properties, it cannot embed out-of-sample points without recomputing the graph Laplacian and its eigenvectors (11). The idea of t-SNE is originally from the method of Stochastic Neighbor Embedding (SNE), in which it converts the high-dimensional Euclidean distances into low-dimensional Euclidean distances using Gaussian kernels. However, SNE has two major flaws. The non-symmetric conditional probability matrices result in a complicated gradient formula, and Gaussian kernels have so-called "crowding problem" in the low-dimensional space. t-SNE employs a heavy-tailed distribution in the low-dimensional space to reduce the optimization complexity and the crowding problem (3).

We finally apply both methods to a real high-dimensional dataset derived from breast cancer research and compare the results. The goal of the breast cancer research is to differentiate two types of cells – breast cancer cells and breast cancer stem cells. This differentiation is based on the observed changes of genome structure and function. We will also identify the regions of change with genes

*Department of Mathematics and Department of Computational Medicine & Bioinformatics, Medical School, University of Michigan

†Department of Mathematics, University of Michigan

‡Department of Electrical and Computer Engineering, University of Michigan

§Department of Computer Science & Engineering, University of Michigan

corresponding to those regions. All information of genome structure and function are condensed into a rectangular feature matrix, in which each row stands for 1 million base pairs (1 Mb resolution) of the DNA sequence. The first four columns represent network centrality measures that capture genome structure and the fifth column represents gene expression, which captures genome function. For example, chromosome 14, which is approximately 89 million base pairs long (after removal of unmappable regions), has the feature matrix of size 89×5. We also include some analysis using PCA on the same dataset for comparison purpose.

This work is organized in six sections. In Section two, we detail the Laplacian Eigenmaps method. In Section three, we first introduce the method of Stochastic Neighbor Embedding (SNE) which is considered as the foundation for t-SNE and explore t-SNE with two significant advantages compared to SNE. Implementations of the two methods and an application to the breast cancer research are presented in Section four. We conclude with several novel ideas in Section five, and detail each team member's contributions in Section six.

2 Laplacian Eigenmaps

The Laplacian Eigenmaps technique is a non-linear dimensionality reduction method proposed by Mikhail Belkin and Partha Niyogi in 2001. The technique fundamentally assumes the data lies in some low-dimensional manifold $\mathbb{R}^m$ embedded in a high-dimensional space $\mathbb{R}^d$ ($m < d$). The method constructs a weighted graph of the data and minimizes an objective mapping function defined on the Laplacian of the graph, which is considered an approximation to the low-dimensional manifold. Because of the geometric approach of Laplacian Eigenmaps, the method has locality preserving properties.

Consider a data set of points $\mathbf{x}_1, ..., \mathbf{x}_n$ in $\mathbb{R}^d$. Each data point will be a node in the eventual graph. The first step in constructing the graph is forming the adjacency matrix of the data set. We place place an edge (i.e. connection) between two nodes i and j if the data points $\mathbf{x}_i$ and $\mathbf{x}_j$ are close to each other. Closeness can either be defined by Euclidean distance or by a k-nearest neighbors approach.

For Euclidean distance, nodes i and j are connected if

$$||\mathbf{x}_i - \mathbf{x}_j||^2 < \epsilon \tag{1}$$

for some $\epsilon > 0$. This approach is very geometrically intuitive but requires tuning the hyperparameter ϵ and often results in graphs with several connected components.

In the latter approach, nodes i and j are connected if i is one of the k-nearest (Euclidean distance) neighbors of j or j is one of the k-nearest neighbors neighbors of i. While the tuning of k is simpler than that of ϵ, this method is less geometrically intuitive.

The next step is to assign weights to connections between nodes. Similarly to the first step, there are two methods for weighting edges (simple and Heat Kernel).

In the first method, assign a weight of 1 between nodes i and j if they are connected and a weight of 0 if the nodes are not connected.

$$W_{ij} = \begin{cases} 1 & \text{nodes } i,j \text{ connected} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

In the second method, we utilize the heat kernel to assign a weight of

$$W_{ij} = \begin{cases} e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{t}} & \text{nodes } i,j \text{ connected} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

between nodes i and j if they are connected. The Laplacian of the graph can be considered an approximation of the Laplace-Beltrami operator $\mathcal{L}$ (divergence of the gradient: $\mathcal{L}f = \nabla \cdot \nabla f$) of the objective function defined on the low-dimensional manifold. Further, the Laplace-Beltrami operator on the manifold is intimately related to heat flow, as the heat equation is the PDE $\frac{\partial f}{\partial t} = \mathcal{L}f$. The solution to the heat equation is the heat kernel, which motivates its use in weighting connections between nodes in the graph.

In the final step of the method, we solve for the eigenvalues and eigenvectors of

$$L\mathbf{v} = \lambda D\mathbf{v} \tag{4}$$

for each connected component of the graph. Here, W is the matrix formed during the second step of the matrix, D is the degree matrix of the graph ($D_{ii} = \sum_j W_{ji}$), and L is the graph Laplacian ($L = D - W$). Let $\mathbf{v}_0, ..., \mathbf{v}_{n-1}$ be the eigenvectors of the above equation in ascending order of their respective eigenvalues. Because the smallest eigenvalue is 0, the smallest eigenvector $\mathbf{v}_0$ is not used in the mapping. Finally, data points $\mathbf{x}_i \in \mathbb{R}^d$ are mapped to the low-dimensional space $\mathbb{R}^m$ ($m < d$) by

$$\mathbf{x}_i \mapsto (\mathbf{v}_1^{(i)}, ..., \mathbf{v}_m^{(i)}) \tag{5}$$

Despite its simplicity, Laplacian Eigenmaps suffers from several limitations. First, the method is very sensitive to how the dataset graph is constructed. For example, the model yields significantly different mappings depending on whether the adjacency matrix is built with an ϵ-neighborhood approach (1) or with a k-nearest neighbors approach. Further, hyperparameters need to be tuned for both forming the adjacency matrix and for assigning weights to the edges (if the heat kernel method is used). Second, the model cannot embed novel data points without re-building the graph and re-solving the generalized eigenvector problem. However, some out-of-sample points can be approximated by exploiting Laplacian Eigenmap's locality preservation properties, and Bengio et. al. proposed extensions to the model to allow for novel data embedding. Finally, it is not known in which empirical contexts the assumed manifold properties (uniform probability density, continuity, etc.) are false, and how to account for these misconceptions.

For hyperparameter tuning on our models, we used a grid search, varying the method for determining the weight matrix between the heat kernel and k-nearest neighbors, as well as varying the gamma hyper parameter used for the heat kernel and the number of neighbors considered in k-NN. Performance of the embedding is quantified by considering the validation accuracy of a binary classifier trained on the low dimensional data. We used a Random Forest Classifier, where hyper parameters for the classifier itself were found by randomized search. We found that calculating the weight matrix by using k-nearest neighbors, where k was normally just a bit larger than 10 was the most frequent configuration for this algorithm.

Algorithm 1: Laplacian Eigenmaps.

Input: Dataset $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, manifold dimensionality m.
Output: Low-dimensional data representation $\mathcal{Y}^{(T)} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\}$.
Build the adjacency graph by connecting nodes according to either (1) or k-nearest neighbors
Add weights to graph edges according to either (2) or (3)
for *each connected component in the graph* **do**
  Solve the generalized eigenvector problem (4)
  Map the input data $\mathbf{x}_i$ to its low-dimensional form $\mathbf{y}_i$ according to (5)
end

3 t-Distributed Stochastic Neighbor Embedding

Stochastic Neighbor Embedding (SNE) is a nonlinear learning method proposed by Hinton and Roweis (4) in 2002 that is capable of preserving neighbor identities. The key idea of SNE is using conditional probability to represent the similarity between two data points in a high-dimensional space, and modeling the similarity onto low-dimensional counterparts. The Gaussian kernel becomes naturally suited to SNE, where for a data point x_i, the neighbors of x_i have high conditional probability $p_{j|x}$, whereas the separated data points have infinitesimally small values of $p_{j|x}$. We can write down the conditional probability $p_{j|i}$ using Bayes rule.

$$p_{j|i} = \frac{p_{j,i}}{\sum_{k \neq i} p_{k,i}} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\boldsymbol{\sigma}_i^2\right)}{\sum_{k \neq i} \exp\left(-\|x_i - x_k\|^2 / 2\boldsymbol{\sigma}_i^2\right)} \tag{6}$$

The Gaussian parameter $\boldsymbol{\sigma}_i$ plays an important role here because it determines the similarities in the high-dimensional space. Hinton and Roweis (4) propose a binary search for $\boldsymbol{\sigma}_i$ using the idea of the Shannon entropy. Define the perplexity as

$$Perp(P_i) = 2^{H(P_i)} \tag{7}$$

where P_i is the conditional probability distribution over all other data points given x_i and $H(P_i)$ is the Shannon entropy given by

$$H(P_i) = -\sum_j p_{j|i} \log_2 p_{j|i} \tag{8}$$

The perplexity usually is chosen between 5 and 50. It can be imaginable that the entropy of P_i increases as $\boldsymbol{\sigma}_i$ increases, and one may use information of the valid number of neighbors of x_i to determine the value of $\boldsymbol{\sigma}_i$. On the other hand, the similarity between low-dimensional counterparts y_i and y_j is defined analogously.

$$q_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2\right)}{\sum_{k \neq i} \exp\left(-\|x_i - x_k\|^2\right)} \tag{9}$$

where the variance of the low-dimensional Gaussian is fixed to be $\frac{1}{\sqrt{2}}$. Although the same variance may lose some data properties in the low-dimensional space, it is sufficient and convenient to exhibit the local architectures from the high-dimensional data.

Eventually, the cost function is represented by a sum of Kullback-Leibler divergence, a commonly used tool that measures the difference between two probability distributions.

$$C = \sum_i KL(P_i|Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}} \tag{10}$$

where Q_i is the conditional probability distribution over all other mapping points given y_i. The minimization of C can be accomplished by using a gradient descent method.

There are two main disadvantages of SNE. First, the cost function is complicated and so is its gradient. More importantly, SNE has the so-called "crowding problem." Due to the nature of Gaussian kernel, the moderately distance datapoints will not be nearly large enough compared with nearby datapoints in low-dimensional space, which would lead to inaccurate distances between datapoints. In order to overcome the first problem, Maaten et al. (3) propose to symmetrize these two conditional probabilities $p_{j|i}$ and $q_{j|i}$ by the following:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n},\ q_{ij} = \frac{\exp\left(-\|y_i - y_j\|^2\right)}{\sum_{k \neq l} \exp\left(-\|y_k - y_l\|^2\right)} \tag{11}$$

where q_{ij} is the joint probabilities. On the other hand, the second issue can be alleviated by using a Student t-distribution with one degree of freedom among low-dimensional counterparts, which means q_{ij} will be modifies to

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|y_k - y_l\|^2)^{-1}} \tag{12}$$

Therefore, the cost function and its gradient are given by

$$C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{13}$$

$$\frac{\partial C}{\partial y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)(1 + \|y_i - y_j\|^2)^{-1} \tag{14}$$

respectively. There are many hyperparameters in t-SNE. The most significant one is referred as the perplexity that determines the Gaussian parameter $\boldsymbol{\sigma}_i$ in high-dimensional space. Usually, large perplexity would result in small $\boldsymbol{\sigma}_i$ and subsequent Kullback-Leibler divergence. However, small value of Kullback-Leibler divergence may not work very well in practice. Hence, we narrow the range of perplexity from 25 to 35 and check the minima and performance for each perplexity. We finally decide to use $Perp = 30$ for this project.

Although t-SNE is superior to other nonlinear dimensionality reduction techniques for data visualization, t-SNE has three major weaknesses (3). First, except data visualization, it is not evident that how t-SNE perform on other task of dimensionality reduction such as feature extractions. Second, t-SNE mainly preserves local properties of the data, so it may be sensitive to the "curse of the intrinsic dimensionality" of the data (5). Three, the cost function of t-SNE is not convex, so there is no guarantee that it will converge to the global minimum.

The pseudo code of t-SNE is given by the following, in which the momentum term $\alpha(t)$ is added to the gradient in order to speed up the optimization and avoid local minima.

Algorithm 2: t-Distributed Stochastic Neighbor Embedding.

Input: Dataset $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, Perplexity $Perp$, Number of iterations T, Learning rate η, and Momentum $\alpha(t)$.
Output: Low-dimensional data representation $\mathcal{Y}^{(T)} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\}$.
Compute pairwise affinities $p_{j|i}$ with perplexity $Perp$ using (6)
Set $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$
Initiate $\mathcal{Y}^{(0)}$ from $\mathcal{N}(0, 10^{-4}I)$
for *t=1,2,…,T* **do**
  Compute low-dimensional affinities q_{ij} using (12)
  Compute gradient $\frac{\partial C}{\partial \mathcal{Y}}$ using (14)
  Set $\mathcal{Y}^{(t)} = \mathcal{Y}^{(t-1)} - \eta \frac{\partial C}{\partial \mathcal{Y}} + \alpha(t)(\mathcal{Y}^{(t-1)} - \mathcal{Y}^{(t-2)})$
end

4 Evaluation on Real Data: Differentiating Cancer Cells

It is widely accepted that breast cancer is one of the most difficult diseases among women in today's society. Based on the journal Global Cancer Statistics, over 500 million deaths are estimated from breast cancer across the world (6). Surgery, radiotherapy, and chemotherapy are considered the most common treatments for breast cancer (7). However, these treatments only target the breast cancer cells (BCCs). The patients are vulnerable to have relapse and metastasis because of the presence of breast cancer stem cells (BCSCs), which are cells that possess the ability to both self-renew and differentiate to new BCCs (8). In order to develop clinical therapies to eliminate BCSCs, it is necessary to understand the genomic distinctions in both structure and function between BCCs and BCSCs.

Nowadays, there are many tools that can be used to detect genome structure and function. The chromosomal conformation (Hi-C) map represents the spatial proximity of different parts of genome capturing the genome structure, while the RNA-Seq data exhibits the activity levels at different parts of genome capturing the genome function (9). Liu et al. (10) extract genome structural centrality measures, including degree centrality, closeness centrality, betweeness centrality, and eigenvector centrality from Hi-C data and concatenate with RNA-Seq data to provide a comprehensive genome-wide form-function evolution over time. These centrality measures are able to provide structural information in low dimensions from a network prospective.

A research group at the University of Michigan applied the techniques of Hi-C map and RNA-Seq on BCCs and BSCS, and they obtained structural matrices and functional vectors of 22 chromosomes for BCCs and BCSCs respectively (chromosome X is not considered in the experiment). After centrality analysis and normalization, we obtain the data for this project. There are 22 feature matrices that are composed of the four centrality measures derived from the structural matrices and the functional vectors for BCCs. Also, there are 22 similarly constructed feature matrices for BCSCs. The resolution of our Hi-C and RNA-Seq data in this case is 1Mb, which means each row of the feature matrices represents 1 million base pairs of the DNA sequence.

In order to visualize the data in low dimension and compare the difference between BCCs and BCSCs easily, we apply the dimension reduction techniques - Laplacian Eigenmaps and t-SNE - that we discussed above. For comparison, we also include the method of Principal Component Analysis (PCA). The idea is that we compute the distances among low-dimensional data points to represent the observed changes of genome structure and function and identify the top five largest changed regions with genes corresponding to those regions.

The results are shown in Figures 1, 2 and 3 for selected chromosomes. Laplacian Eigenmaps are able to successfully separate the contributions of BCCs from BCSCs in the global sense for some chromosomes, and t-SNE provides a nice visualization of the high-dimensional data and the subtle local distinctions between BCCs and BCSCs. Nevertheless, PCA fails to provide clear and meaningful interpretations for most of the chromosomes due to the high linearity and dense clustering of its projected data points. Now, we identify the regions with five greatest changes in both structure and

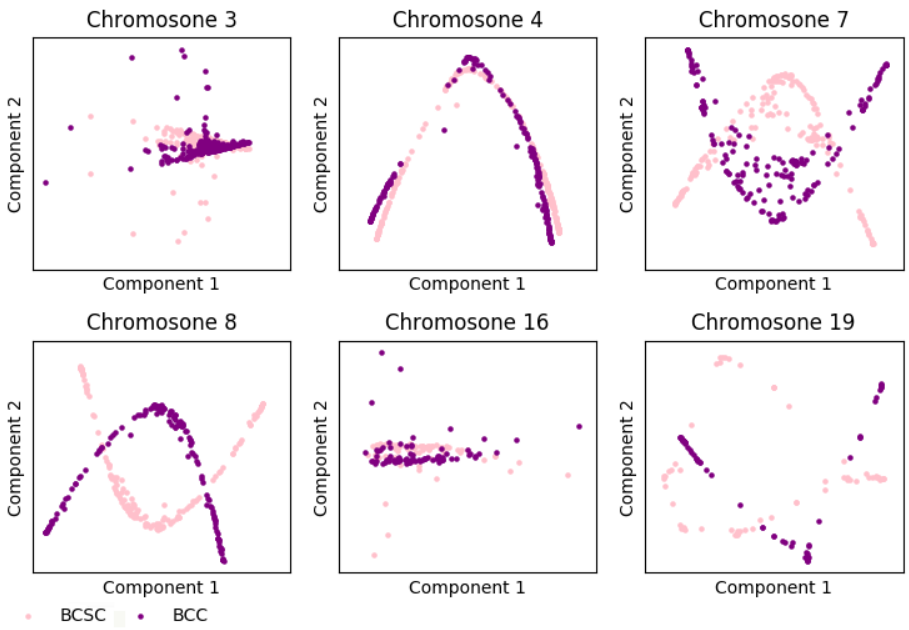

Figure 1: **Dimension reduction of feature matrices through Laplacian Eigenmaps**. The figure shows the low-dimensional visualization of feature matrices derived from the Hi-C and RNA-Seq data for selected Chromosomes using Laplacian Eigenmaps. In Chromosomes 4, 8, 7 and 16, the low-dimensional mapping of the feature matrices clusters together and form two "V" shapes for BCCs and BCSCs, which indicates the structural and functional distinctions between BCCs and BCSCs are small in these Chromosomes. In Chromosome 7 and 8, the low-dimensional mapping is able to separate the contributions from BCCs and BCSCs, which demonstrates the large structural and functional distinctions between BCCs and BCSCs in these Chromosome.

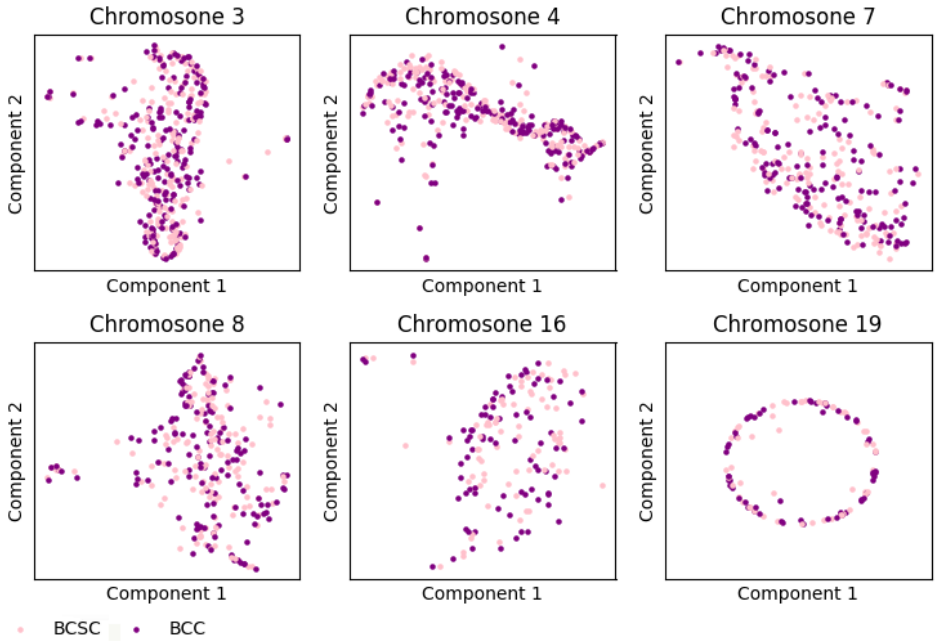

Figure 2: **Dimension reduction of feature matrices through t-SNE.** The figure shows the low-dimensional visualization of feature matrices derived from the Hi-C and RNA-Seq data for selected Chromosomes using t-SNE. In all the chromosomes, the low-dimensional mapping of the feature matrices exhibits visually beautiful and clear patterns of the local properties between BCCs and BCSCs. Furthermore, the low-dimensional configuration implies that the structural and functional distinctions between BCCs and BCSCs are subtle.

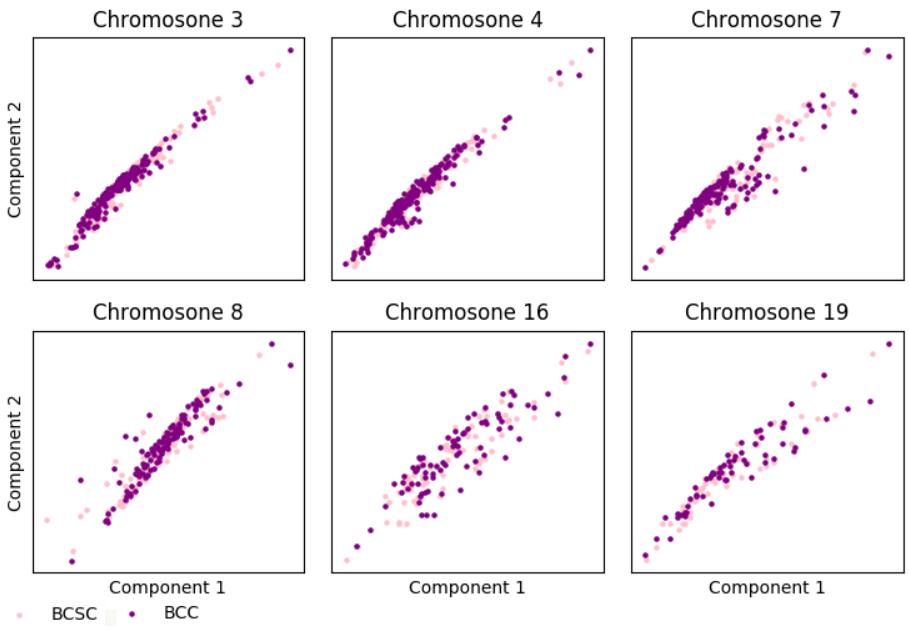

Figure 3: **Dimension reduction of feature matrices through PCA.** The figure shows the low-dimensional visualization of feature matrices derived from the Hi-C and RNA-Seq data for selected Chromosomes using PCA. In all the chromosomes, the low-dimensional mapping of the feature matrices clusters together and presents linearity between the two components. In particular, in Chromosome 3, 4, 7 and 8, the clusters are very dense, and it is not easy to distinguish the differences between BCCs and BCSCs.

function for Chromosome 8, which has 141 million base pairs of the DNA sequences. By computing the distances in the low-dimensional space, Laplacian Eigenmaps gives genomic regions at the 47, 95, 48, 97, and 58 millionth base pairs. t-SNE yields genomic regions at the 107, 31, 42, 48, and 76 millionth base pairs. For example, the overlapping region at 48 millionth base pair contains two genes CYCSP22 and SNTG1. Although we are not able to tell which method is better than the other, Laplacian Eigenmaps and t-SNE both offer the chance to identify genes that may be different in BCCs and BCSCs.

5 Conclusion

In this paper, we introduced the manifold learning algorithms of Laplacian Eigenmaps and t-Distributed Stochastic Neighbor Embedding and evaluated both algorithms on the BCC/BSCS dataset. For comparison, we also performed Principal Component Analysis on the same dataset.

Provided with non-linearly correlated data, both manifold learning techniques have better performances compared to PCA. Laplacian Eigenmaps succeeded best in separating the data the classification, and tended to preserve the high-dimension data's global structure. However, Laplacian Eigenmaps required ample hyperparameter tuning and had significant overlap between classes for some chromosomes. t-SNE succeeded in visualizing the high-dimensional data and preserving its local differences in the data. However, t-SNE was not as successful in separating the BCC data points from the BCSC data points.

A major limitation when comparing manifold learning algorithms is the lack of standard quantitative metrics for that can be applied across different models. For example, PCA is often associated with the percentage of variation explained by its low-dimension projection, but such a metric does not have as direct of an application to Laplacian Eigenmaps or t-SNE. While our parameter sweep metric for Laplacian Eigenmaps, in which the model is evaluated by the performance of a random forest classifier on the low-dimension mapping, produces a relatively-meaningful and quantitative

output, the metric does not definitively describe how well each method projects the data into a low-dimensional subspace.

6 Team Member Contributions

In addition to having distinct primary focuses, each author participated in all endeavors necessary to complete this project. All team members were dedicated and highly collaborative, and there were not significant imbalances in effort. Can Chen focused on processing the BSCS data and writing the report, specifically for sections relating to t-SNE and application on breast cancer cells. Yuting Liu focused on implementing the t-SNE algorithm, which took more time than expected to cover tiny details and to speed it up. Peter Paquet focused on writing and editing the report, specifically for sections relating to Laplacian Eigenmaps. James Connolly focused on implementing the LEM algorithm, and spent most of his time figuring out how to correctly calculate the Laplacian matrix and writing the hyperparameter sweep.

References

[1] D. Chen and H. G. Muller, *Nonlinear manifold representations for functional data*, The Annals of Statistics, 40(1) (2012), pp. 1–29.

[2] M. Belkin and P. Niyogi, *Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering*, Advances in Neural Information Processing Systems 14, MIT Press, 2002, pp. 585–591.

[3] L. Maaten and G. Hinton, *Visualizing Data using t-SNE*, Journal of Machine Learning Research 9 (2008), pp. 2579–2605

[4] G. Hinton and S. Roweis, *Stochastic Neighbor Embedding*, Advances in Neural Information Processing Systems 15,MIT Press, pp. 857–864, 2003

[5] Y. Bengio, *Learning deep architectures for AI*, Technical Report 1312, Universite de Montreal, 2007.

[6] L. A. Torre,F. Bray, and R. L. Siegel, *Global Cancer Statistics*, CA: A Cancer Journal for Clinicians (65)(2015) 87–108.

[7] R. Siegel, J. Ma,Z. Zou, and A. Jemal, *Cancer Statistics*, CA: A Cancer Journal for Clinicians (64)(2014) 9–29.

[8] C. Anders, L. A. Carey, *Understanding and treating triple negative breast cancer*, Oncology (22)(2008), 1233–1239.

[9] E. Lieberman-Aiden, N. L. van Berkum, L. Williams, M. Imakaev, T. Ragoczy, A. Telling, I. Amit, B. R. Lajoie, P. J. Sabo, M. O. Dorschner, R. Sandstrom, B. Bernstein, M. A. Bender, M. Groudine, A. Gnirke, J. Stamatoyannopoulos, L. A. Mirny, E. S. Lander, and J. Dekker, *Comprehensive Mapping of Long-Range Interactions Reveals Folding Principles of the Human Genome*, Science, 326(5950) (2009), pp. 289–293.

[10] S. Liu, H. Chen, S. Ronquist, L. Seaman, N. Ceglia,W. Meixner, P. Chen, G. Higgins, P. Baldi, S. Smale, A. Hero, L. A. Muir, and I. Rajapakse, *Genome Architecture Mediates Transcriptional Control of Human Myogenic Reprogramming*, iScience (2018).

[11] Y. Bengio, J. Paiement, P. Vincent, N. Le Roux, M. Ouimet, *Out-of-Sample Extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering*, Advances in Neural Information Processing Systems (2004).

7 Supplementary

The code associated with this project can be found at https://github.com/conjam/545RefRepo. The dataset used in the report is confidential. For inquiries regarding the dataset, please contact the authors of this work.

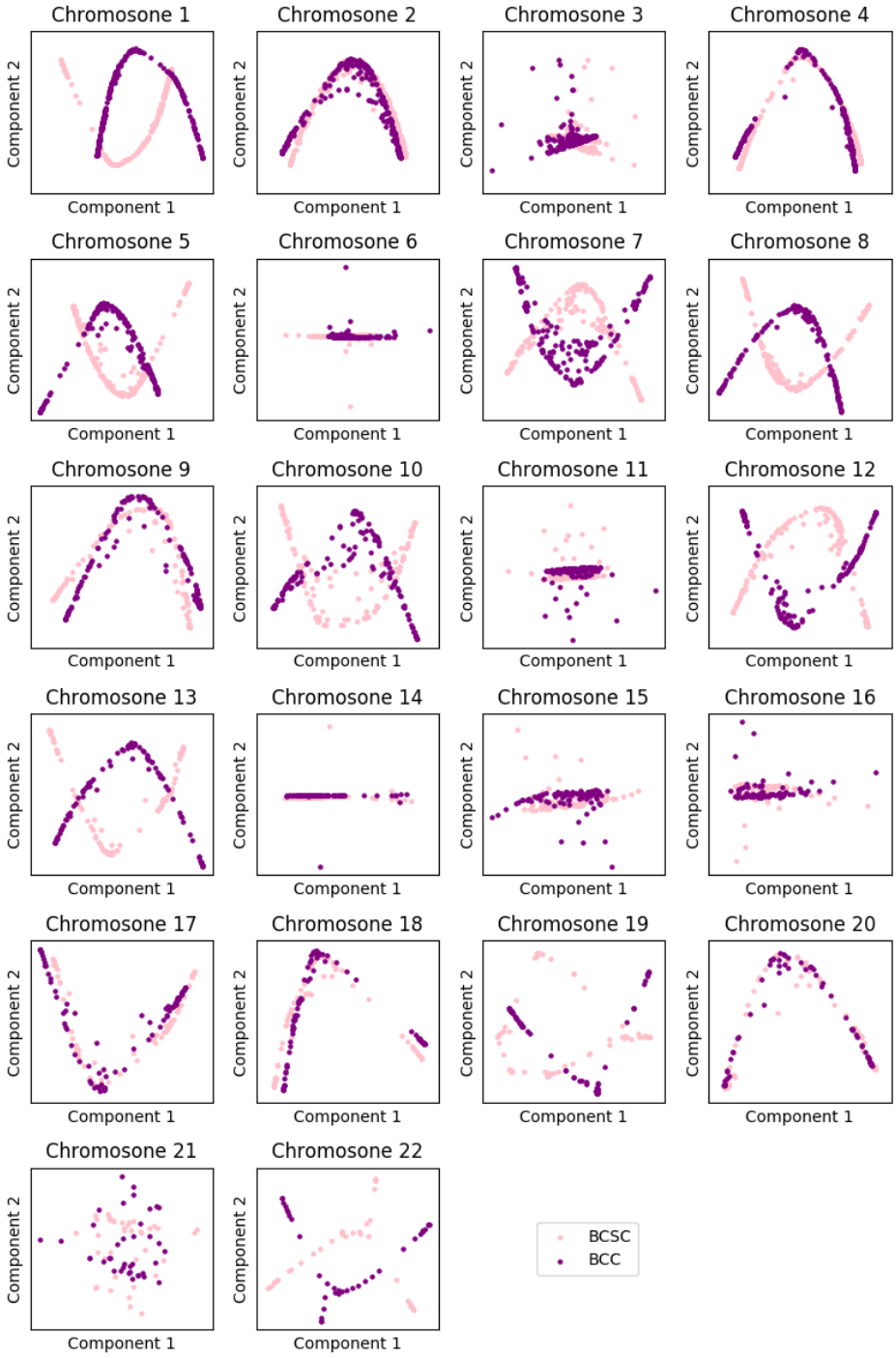

Figure 4: **Dimension reduction of the feature matrices using LEM for all 22 chromosomes.**

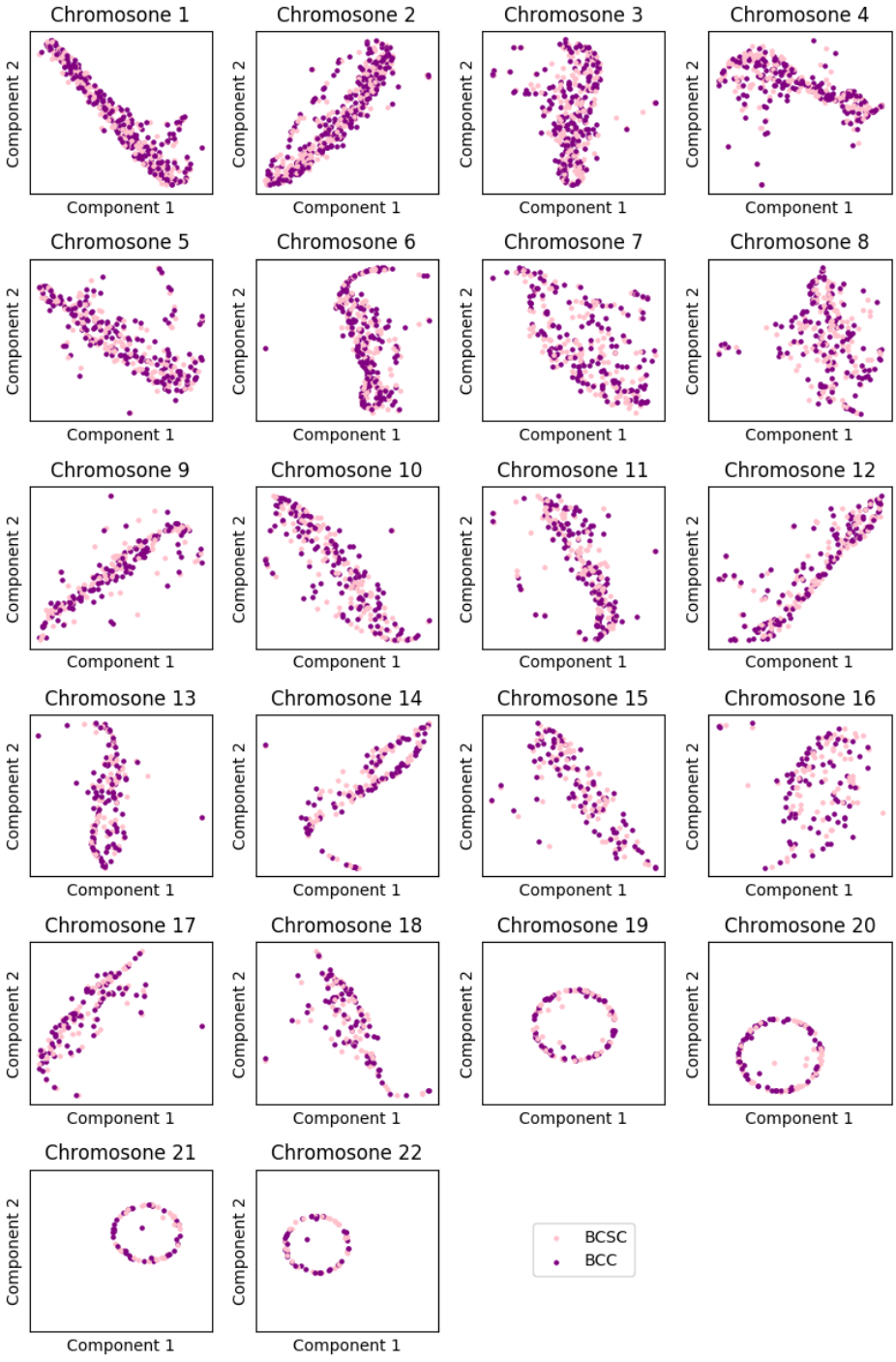

Figure 5: **Dimension reduction of feature matrices through t-SNE for all 22 chromosomes.**