UNIVERSITY OF AMSTERDAM

MASTER'S THESIS

Diffuse Intrinsic Pontine Glioma and the Application of Multi-Modal Data Fusion Techniques

Author:
Maximilian LOMBARDO

Supervisor:
Dr. Shi YU

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

IBM Center for Advanced Studies
Computational Science

November 7, 2018

Declaration of Authorship

I, Maximilian LOMBARDO, declare that this thesis titled, "Diffuse Intrinsic Pontine Glioma and the Application of Multi-Modal Data Fusion Techniques" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

"This is a quote."

Maximilian Lombardo

UNIVERSITY OF AMSTERDAM

Abstract

Computional Science

Computational Science

Master of Science

Diffuse Intrinsic Pontine Glioma and the Application of Multi-Modal Data Fusion Techniques

by Maximilian LOMBARDO

Large amounts of data are quickly becoming available in many scientific fields, allowing for the utilization of a variety of techniques that help us learn patterns from this data. However, with this newly available wealth of data, scientific practitioners face new challenges in making sense of it all. These challenges are particularly present in the medical and biological disciplines where new technologies have allowed for the rapid accumulation of relevant patient and molecular data.

Along with developing methods that can deal with the increased scale of new data types, being able to make decisions which take into account many different data sources is also a unique challenge which needs to be dealt with. A prime example might be the analysis of sequencing data to make diagnosis decisions about a particular patient. In a situation where DNA and RNA data is available, it would be ideal to be able to take advantage of all available information since each one of these modalities should not only confirm abnormal changes in the other other types, but may also provide complementary "side" information which would be lost had only one particular data type been taken into consideration. These arguments are even more valid for rare diseases, where sample sizes are usually quite small, making it essential to use all available data to learn a model which can be used for decision making.

Using this motivation, this thesis takes a look at different approaches for data fusion, focusing on methods which perform the combination of modalities in a data dependent manner. In addition to the exploration of established methods, their potential applications will be introduced in the context of glioblastoma multiforme and diffuse intrinsic pontine glioma.

Acknowledgements

I would like to thank Dr. Shi Yu, Dr. Zoltan Zlavik, and Dr. Jaap Kaandorp.

Contents

List of Figures

List of Tables

List of Abbreviations

LAH	**L**ist **A**bbreviations **H**ere
WSF	**W**hat (it) **S**tands **F**or

For my parents

Chapter 1

Data Fusion Methods

1.1 Multi-modal Learning Motivation

Many natural systems can be described by different measurements which represent different perspectives of understanding with respect to how that system operates. Since we are able to leverage a single data perspective for the learning and prediction of properties and outcomes of that system, then it is logical to conclude that it is possible to leverage multiple perspectives to build models that can learn more efficiently and with higher accuracy than when using a single perspective alone. In this writing, we shall focus on specific scenarios which meet the requirements for multimodal learning, both with respect to the data types we might consider and the learning frameworks we might employ. In particular we will be interested in the application of multimodal learning techniques to biological systems, and more specifically to learning the behavior and distinguishing characteristics of disease.

1.2 Multimodal Data Types

Multimodal data can broadly be described as any datasets that provide complementary information about a specific subject (in our case, a biological sample). Li et. al. have previously provided a breakdown of four scenarios where multimodal learning techniques could be leveraged [1]:

- **Scenario 1:** We possess measurements of different groups (classes) with the same feature sets.
- **Scenario 2:** Our sample set remains consistent, but we have multiple distinct feature sets with which to describe each of the samples.
- **Scenario 3:** A more classic experimental setup where we have the same sample set and feature set under different experimental conditions.
- **Scenario 4:** Finally, the scenario where we have different sample sets and different feature sets, but under the examination of the same system, a scenario where our goal is to try and learn relations between the multiple independently obtained data sets.

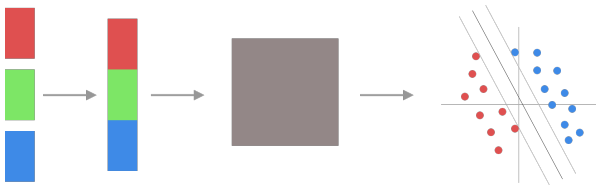
[1] A review on machine learning principles for multi-view biological data integration

Scenarios two and four are the most interesting to leverage and in the context of biological study are also referred to as multi-omics data. In this thesis, we will specifically be looking a few of these scenarios, but focus on the situations where we have multiple, completely distinct feature sets which describe the same set of samples. In the biological context it makes theoretical sense to leverage this data type because these data types often are dependent upon each other. For example, proteins are constructed based on a template called an RNA transcript, which is in turn based on a template called DNA. An aberration in any one of steps can result in the absence of a particularly crucial protein or in the introduction of mistakes in the protein which can affect its ability to function. The propagation of such a change through these sequential templates can of course result in a disease state within a cell or individual. In addition to mistakes being propagated throughout the system, it is also possible that changes in any of these templates directly contributes to generation of a disease state. This might manifest itself in changes in each of the templates which impact different parts of a particularly important pathway. In this scenario, it would be important to know if multiple detrimental changes occurred such that correction of one disease causing change would not be sufficient to correct for the disease condition.

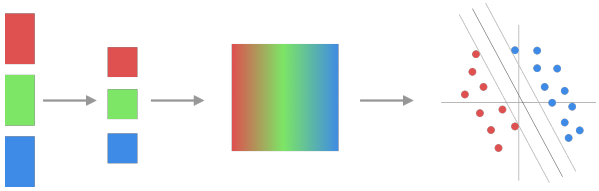
1.3 Data Fusion Broad Approaches

Data fusion is the process by which different data modalities are integrated in an effort to build more informed predictive models for different learning tasks. Data fusion can be classified into three different groups: early, intermediate, and late fusion.[2]

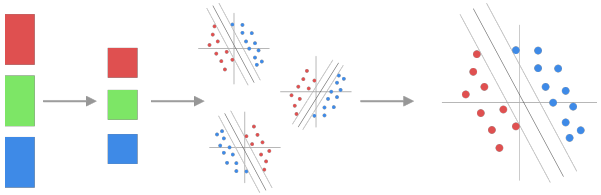
[2] source needed



(A) Early Fusion



(B) Intermediate Fusion



(C) Late Fusion

FIGURE 1.1: General approaches to data fusion: Early integration simply involves the concatenation of features from different data sources and training a algorithm on that "new" dataset. Intermediate integration constructs a learning machine from all available data sources. Late fusion is essentially an ensemble method, where multiple learners on are trained and the resulting predictions are combined

- **Early Fusion:** All features from individual modalities are concatenated into a single feature representation, whose size can be managed through a dimensionality reduction process.
- **Intermediate Fusion:** Casts all data types into a common format (graph, kernel, etc.) so that combination occurs on the common representation.
- **Late Fusion:** Individual predictive models are built on each data type independently and combined after predictions have been (class assignments or regression target values) made. Combination typically occurs in a voting scheme for classification tasks and an averaging scheme for regression tasks.

While all three of these approaches have been shown to be valid for the data fusion problem, intermediate methods may be the most interesting because of their ability to take all data sources into account during a single learning process and their ability to typically achieve higher performance than simpler formulations of the problem.[3] Moreover, the ability to learn intrinsic structure between samples using all available data sources simultaneously is desirable characteristic of this approach.

There are several different approaches that have been utilized in the integration of omic data which we will introduce briefly. These include matrix factorization methods, bayesian based approaches, correlation based analysis, multiple kernel learning Methods, and network fusion methods. In the following two chapters we will take a more in depth look at the mathematical basis behind multiple kernel learning and network fusion given that these frameworks seem to be the most popular for dealing

[3] source needed

with multiomics problems. Following an explanation of the math behind some unsupervised and supervised versions of these two approaches, we will examine their performance on a variety of multiomics data sets and finally discuss their suitability for brain cancers and rare disease.

1.4 Approaches for Intermediate Data Fusion

The general approach of most intermediate fusion problems considers relationships between samples in order to cast different data types into a common format. The first two methods (matrix factorization and correlation based analysis) achieve this by projecting each data modality onto a shared subspace. Bayesian methods use a bayesian framework to assess the probability of specific state given each condition and utilize the joint probability of these posterior predictions to perform the integration itself. Finally, kernel based methods and graphical models express each individual dataset in terms of the similarity between individual samples. The similarities derived from each dataset are then used to arrive at a final similarity from which relationships, classifications, and regression values can be determined. [4]. This holistic approach to learning similarity can also be applied to problems with a single data modality by combing different similarity metrics which have been calculated on the same dataset in order to learn an optimal similarity.

1.4.1 Matrix Factorization

Matrix factorization for data integration has the aim of finding a shared subspace between the different modalities, which acts as a low dimensional shared feature matrix (shared across all the different data sets). [5]. This method generally works by optimizing the following objective function:

$$minimize: \sum_{i=1}^{M} ||X_i - WH_i||_2^2$$

$$subject\ to: W \geq 0,\ H \geq 0$$

Where M is the number of data modalities that are to be integrated, X_i is one of the original datasets, H_i is the data specific coefficient matrix, and W is the common basis that these data are being projected onto. The goal here is to find a factorization which minimizes the reconstruction error of the system. After this representation has been found, the use of classical machine learning methods can be performed on the W matrix (i.e. perform clustering in the lower dimensional, joint latent subspace). As Wang et.al. point out, these approaches typically require non-convex optimization and are not robust to large scale data. [6]. Additonally, these methods are difficult interpret once the data have been projected into the latent subspace and therefore offer little opportunity to gain biological insight from this learning process. Other popular matrix factorization based approaches to the data integration

[4] More Is Better: Recent Progress in Multi-Omics Data Integration Methods

[5] A novel computational framework for simultaneous integration of multiple types of genomic data to identify microRNA-gene regulatory modules

[6] Integrative clustering methods of multi-omics data for molecule-based cancer classifications

problem in biological settings include a method called iCluster, which also assumes that original datasets can be described in a joint latent subspace, but relaxes the non-negative constraints of the problem. Additionally, iCluster makes use of a noise term in its reconstructed version of the original matrix. [7]

1.4.2 Correlation Based Analysis

Canonical correlation Analysis is the classical approach to many data integration problems and involves finding projections of each individual dataset such that correlation between the canonical variables (data set specific latent variables which are constructed via linear combinations of the variables pertaining to a specific dataset) is maximized. Additionally, constraints are placed on this problem such that the correlation between any different pairs of canonical variables is 0 so that we essentially end up with dataset specific projections to a subspace that has a reduced dimensionality, and where each dimension of the subspace summarizes co-occurrence of variables such that dimensions do not provide redundant information. Variables from a dataset (or datasets) that map to the same dimension in the lower dimensional subspaces are assumed to be highly correlated and can be combined into meta-features or can be thought of describing a similar topic or a higher level representation than individual variables. What this means in practice is that sets of features are extracted from each individual dataset which correspond to dimensions in the subspace, thus reducing the dimensionality in both datasets and also giving us two sets of features which are maximally correlated with each other. This is typically performed on two datasets such that if we consider the two projections $u = a^t x \; and \; v = b^t y$ for the two sets of variables $x \; and \; y$, then we can express the objective function as such:

$$maximize: \frac{E[uv]}{\sqrt{E[u^2]E[v^2]}}$$

1.4.3 Bayesian Methods

Bayesian methods offer the advantage of being able to make assumptions on the distributions of each individual datasets and that they also offer an intuitive way of combining different datasets together. One such approach is to predict the likelihood of an outcome $L(Y)$ given a specific dataset D_i as compared to seeing that dataset for any other outcome Z given the same dataset (a framework which lends itself most easily towards distinguishing between two binary states)[8]:

$$L(Y|D_i) = \frac{P(D_i|Y)}{P(D_i|Z)}$$

Using this framework for an individual dataset, we can use individual likelihood ratios to arrive at a "joint" likelihood ratio, by calculating the product of all likelihood ratios:

[7] Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis

[8] Madhukar BANDIT

$$L(Y|D_{1,\ldots,M}) = \prod_{i=1}^{M} L(Y|D_i) = L(Y|D_1)L(Y|D_2)...L(Y|D_M)$$

This specific approach makes the assumption of independence between datasets which may be questionable for multi-omics data given that, different omics modalities should in theory be dependent upon one another.

1.4.4 Kernel Methods and Network based methods

Finally, we come to our discussion of data integration methods that take advantage of kernel and graph based frameworks. Both of these methods approach the multiomics problem by representing the dataset in terms of similarity between each sample. Kernel methods represent all of these pairwise comparisons in a square Kernel matrix, preserving all of the similarity information from each pair of comparisons, while network based models construct a graph, converting similarities between samples into links (weighted or unweighted) if they fall above a certain threshold. Since links are only constructed between samples above a certain threshold, potentially useful information is discarded. On the other hand, since networks provide links between nodes, it is possible to capture indirect relationships between samples whereas a kernel based approach only allows for direct comparison. Differences also arise in the way that kernels and graphs are combined from different data modalities. The properties of kernel functions allow for very intuitive combination of different kernel matrices (namely through linear combinations), whereas the combination of multiple graphs requires more involved schemes, although some network based methods can capitalize on the linear combination scheme that we see in kernel based methods. How these combinations occur for each approach will be covered in the next two chapters of this thesis. While we will examine these two approaches as being distinct from the other data integration methods that we described previously, it is important to note that both kernel and network based learning schemes can be adjusted to incorporate approaches from the previously discussed frameworks.

Chapter 2

Kernel Based Methods

2.1 A short introduction to kernel methods

Kernel methods make use of a kernel function which always takes the form of a dot product between two sample vectors (and usually involves some other modification beyond taking just the dot product). Pairwise combinations of each sample are then transformed by the kernel function into a scalar value which can usually be thought of as a similarity metric. These similarity measures then make up the entries of a square, symmetric, positive semi-definite matrix. The main innovation of kernel methods is the ability to implicitly map data points to a higher dimensional feature space, by expressing the relationship between two samples in that higher dimensional space without explicitly computing each sample's representation in higher dimensions. A kernel function can be represented in a general form as such:

$$K(x, x') = \langle \phi(x)\phi(x') \rangle$$

Where K represents the kernel function. Transformation into a higher dimensional feature space can either be explicit or implied, such as in a gaussian kernel.[1]

Furthermore, kernel formulations have been discovered which are able to express the relationships found in text, string, graph, and tree based representations of samples, allowing for the incorporation of non-standard data types into a kernel based framework. Kernel based methods are especially interesting for genomic applications because of their ability to deal with high dimensional data and non-linear relationships between features and the target variable.[2] Furthermore, kernel based formulations exist for many of the tasks which are interesting in the machine learning world, mainly classification, clustering, regression, dimension reduction, and feature discovery. [3]

[1] source needed – CALtech
[2] Kernel Methods for large-Scale genomic Data Analysis
[3] Kernel Methods for Pattern Analysis

TABLE 2.1: Mathematical formulations of some common kernel functions.

Kernel	**Formulation**	**Application**
Linear	$K(x, x') = xx'$	General purpose
Polynomial	$K(x, x') = (\gamma xx' + r)^d, \gamma > 0$	General purpose
Gaussian RBF	$K(x, x') = exp(-\frac{\|\|x-x'\|\|^2}{2\sigma^2})$	General purpose
Sigmoid	$K(x, x') = tanh(\gamma xx' + r)$	General purpose
Need	Mor	Kernels

2.2 Multiple Kernel Learning

The multiple kernel learning (MKL) problem was first formulated by Lanckriett and colleagues[4] and was originally worked into the support vector machine learning (SVM) scheme. Because of the SVM based origin of the MKL problem, the approach is based on convex optimization. The work of Lanckriett takes advantage of the fact the kernel functions and kernel matrices can be mathematically combined (addition, multiplication) while still maintaining the positive semi-definite characteristic needed for the kernel and kernel function to be valid. Linear combinations are typical and in the scheme developed by Lanckriet, weights are learned for each kernel representing a different data source (a development on unweighted or equally weighted combinations of kernel matrices[5]). This is done by adopting a semi-definite programming (SDP) framework. SDP is focused on optimizing convex functions over the convex cone of symmetric, positive-semidefinite matrices. Since kernel matrices are positive semi-definite matrices and are also symmetric, the solution space which is considered in SDP can be considered a search space for an optimal kernel (and furthermore for an optimal kernel resulting from the convex combination of multiple kernels). Here we consider how a particular kernel based algorithm, support vector machine (SVM) can be adapted to support the fusion of multiple kernels.

[4]Lanckriett et. al., 2004 Kernel based integration of genomic data using SDP
[5]Pavlidis et. al 2001 see Lanckriet's references

2.2.1 Support Vector Machines with single and multiple kernels

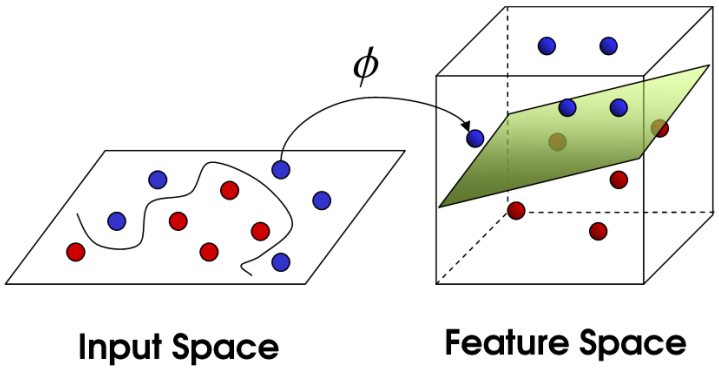


FIGURE 2.1: Using the kernel trick to access a higher dimensional feature space: By using a kernel to train a classifier based features of the data, rather than the original data, we are able to find a representation of the data which is separable by a plane in a higher dimensional space.

The typical motivation for the support vector machine in the binary classification setting is that there are many different solutions to the problem of finding a linear boundary which distinguishes two classes from one another. Let us define the problem such that we have training samples x_i and training labels y_i. Let the class labels $y_i \in -1, 1$ and the sample features $x_i \in R^d$. A solution for distinguishing between these two classes would take the form:

$$f(x) = w^t x - b$$

Where $f(x) > 0 \; \forall \; y_i = 1$, and $f(x) < 0 \; \forall \; y_i = -1$. Classification would occur according to the rule $y_{test} = sign(f(x_{test}))$. The problem with this solution is that it is not unique and that there exist many such hyperplanes which could satisfy the problem. One could imagine that one such solution would be located at the interface of all the positive examples, however this classifier would not deal well with potential noise in a new test case. The same could be said for a classifier that sits at the interface of the negative training examples. To overcome this, we define planes that sit at the interface of each class boundary, and define the linear classifier as being equidistant between these two planes. In this context, these planes are called the support hyperplanes and are based on the training examples that are closest to the classification boundary. They are defined as such:

$$w^t x_i - b \leq -1 \; \forall \; y_i = -1$$

$$w^t x_i - b \geq 1 \; \forall \; y_i = 1$$

Which can also be written as one equation:

$$y_i(w^t x_i - b) - 1 \geq 0$$

In this setting we are interested in defining these two hyperplanes so that we can maximize the distance between them. The distance between a support hyperplane and the separating plane is equal to $\frac{1}{||w||_2}$, and hence the distance between the two support hyperplanes is equal to $\frac{2}{||w||_2}$. Therefore, we will need to maximize this quantity, while maintaining the constraints set by the linear equations that define each support plane, namely that training samples from different classes fall on the the appropriate sides of their respective support planes. The optimization problem in question can be described in the following primal formulation:

$$minimize : \frac{1}{2}||w||_2$$

$$subject\ to : y_i(w^t x_i - b) - 1 \geq 0,\ \forall\ i$$

In order to solve this problem, we first convert it to its dual form using a Lagrangian function:

$$\mathcal{L}(w, b, a) = \frac{1}{2}||w||_2 - \sum_{i=1}^{N} \alpha[y_i(w^t x_i - b) - 1 \geq 1]$$

Because the original optimization was quadratic (and thus convex) we are able to interchange our maximization objective with a minimization objective. Then, to minimize the objective, we need to minimize w, while maximizing α. This optimization then becomes a saddle point problem[6]. We can then find the conditions on w which must hold. This is done by taking the derivatives with respect to w and b and setting them equal to 0. This results in the following solutions:

$$w^* = \sum_i \alpha_i y_i x_i$$

$$b^* = 0 = \sum_i \alpha_i y_i$$

We can then insert these solutions back into the Lagrangian and obtain:

$$maximize : \mathcal{L}_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{ij} \alpha_i \alpha_j y_i y_j x^t{}_i x^t{}_j$$

$$subject\ to : \sum_i \alpha_i y_i = 0$$

$$\alpha_i \geq 0,\ \forall\ i$$

The most important point of this formulation is that it is dependent upon the inner product of two samples and this where we can make use of a kernel function. Furthermore, since we have solved for w using the inner product formulation, we can substitute this representation in our original function of interest, f(x):

[6] https://en.wikipedia.org/wiki/Saddle_point

$$f(x) = \sum_i \alpha_i y_i x_j{}^t x_i - b$$

$$b^* = \sum_i \alpha_j y_j x_j{}^t x_i - y_i$$

In this way, we can make use of a kernel function to determine $f(x)$. This allows us to use kernels which can project the original data points into a higher dimensional space. In the case of the Gaussian kernel, this high dimensional space is implied and is thus not even explicitly calculated, allowing us to leverage the power of a complex feature space without paying the computational price by explicitly calculating a distance in that feature space. Moreover, going back to the expression of $f(x)$ which leverages the inner product formulation, we only need to calculate the inner product for samples which have a corresponding dual variable α which is not equal to 0. The only cases where this scenario arises are when the samples lie on the support hyperplanes and thus the dual variable is typically very sparse, providing another computational benefit when predicting the class of an unseen test case. While, we have presented the linearly separable case here, more complex SVM formulations exist for dealing with non-linearly separable class boundaries and for the allowance of the misclassification of examples. This will not be explained in detail here, but is presented elsewhere[7]. A brief explanation of the non-separable case and how it corresponds to the separable case can be seen as we start with the corresponding primal objective function:

$$minimize(w, b, \zeta) : \frac{1}{2} w^t w + C \sum_{i=1}^{N} \zeta_i$$

$$subject\ to : y_i[w^t \phi(x^i) + b] \geq 1 - \zeta_i$$

$$i = 1, ..., N$$

$$\zeta_i \geq 0$$

In this formulation, ζ and C correspond to slack variables and a regularization parameter respectively. The purpose of the slack variable is to allow for some data samples to be able to cross the support boundary. C is a positive regularization parameter which is meant to control how much "slack" is allowed within the system. The corresponding dual formulation to this primal is:

$$minimize(\alpha) : \frac{1}{2} \alpha^t Y K Y \alpha - \alpha^t 1$$

$$subject\ to : (Y\alpha)^t 1 = 0$$

$$0 \leq \alpha_i \leq C$$

$$i = 1, ..., N$$

[7] http://www.ics.uci.edu/~welling/classnotes/papers_class/SVM.pdf

The above formulation is obviously limited for our purposes because it only takes into consideration one kernel K, representing one data type. To incorporate multiple kernels, the following formulation was proposed by Lanckriett et. Al. and Bach et. Al.[89]:

$$minimize(t, \alpha) : \frac{1}{2}t - \alpha^t 1$$

$$subject\ to : \alpha^t Y K_j Y \alpha,\ j = 1, ..., p$$

$$(Y\alpha)^t 1 = 0$$

$$0 \leq \alpha_i \leq C,\ i = 1, ..., N$$

In this formulation, p refers to the number of kernels. This equation optimizes the $L_\infty - norm$ of the set of quadratic kernel terms. This formulation leads to a sparse solution in α. Alternatively, we can also optimize the $L_2 - norm$ on the set of quadratic kernel terms as shown in Yu et. al.[10]:

$$minimize(t, \alpha) : \frac{1}{2}t - \alpha^t 1$$

$$subject\ to : t \leq ||\gamma||_2$$

$$(Y\alpha)^t 1 = 0$$

$$0 \leq \alpha_i \leq C_i,\ i = 1, ..., N$$

Where $\gamma = \alpha^t Y K_1 Y \alpha, ..., \alpha^t Y K_p Y \alpha, \gamma \in \mathbb{R}^p$. These two formulations based on different norms correspond to solution vectors where the kernel weights accumulate on a few particularly informative kernels or are more evenly dispersed amongst all the kernels respectively. It is therefore easy to see that in the first formulation we are interested in kernel selection whereas the second formulation can be used in cases where we are interested in leveraging all available data sources to come to an optimal solution.

2.2.2 K-Means clustering and Multiple Kernel Learning

While many multiple kernel formulations have been applied to supervised learning algorithms such as the support vector machine described earlier, the framework has also been applied to unsupervised learning algorithms such as the K-means clustering algorithm. In this section, we will explain the k-means algorithm, how it can be converted into a kernelized form of the algorithm, and how that in turn can be used to arrive at a kernel k-means algorithm which allows for the use of multiple kernels, i.e. multiple kernel k-means. The derivations for the Kernel K-means and Multiple Kernel K-means algorithms are adapted from the work of Welling and Gonen, respectively[11][12].

[8] Learning the Kernel Matrix with Semi-Definite Programming
[9] Multiple Kernel learning, Conic Duality, and the SMO Algorithm
[10] L2-norm multiple kernel learning and its application to biomedical data fusion
[11] Welling: Kernel K-Means and Spectral Clustering
[12] Localized Data Fusion for Kernel k-Means Clustering with Application to Cancer Biology

In the K-means algorithm, the goal is to assign points to clusters such that the following objective is minimized:

$$C(z, \mu) = \sum_i ||x_i - \mu_{z_i}||^2$$

Where z_i are assignment variables which can take on the values $z_i = 1, ..., K$, where K is the number of clusters specified at beginning of the algorithm. The variables $\mu_{z_i} = \mu_1, ..., \mu_K$ are used to designate the K cluster centers. In this objective, the goal is to choose a cluster assignment z_i for a point x_i, such that the objective is minimized by assigning the point to the cluster whose cluster center, or cluster mean, μ_k is closed to that point. After assignment of all the points, the cluster centers are re-calculated by averaging the points assigned to that cluster. Iterations of re-assigning the points and re-calculating the cluster centers are then repeated until convergence, when points are not reassigned to new clusters and the cluster centers do not change anymore. The algorithm is started by choosing K starting points, a parameter chosen by the user.

In addition to performing the algorithm in the original space, we can also perform the K-means clustering in feature space, by modifying the objective function as such:

$$C(Z, \mu) = \sum_i ||\phi(x_i) - \mu_{z_i}||^2$$

We can also define a K *by* N assignment matrix, Z_{nk}, which contains a 1 in row K_i, for each column N_i indicating the class membership for that sample. Summing across the rows of that matrix will give us the number of data points assigned to each cluster N_k and we can use this vector to construct the matrix L, where $L = diag[1/N_k]$. Furthermore, we can also define the matrix $\Phi_{in} = \phi_i(x_n)$. Given these definitions, we can also create the matrix M, as such:

$$M = \Phi Z L Z^T$$

Where this matrix is composed of N columns, where each column contains a copy of the cluster mean μ_k of the cluster which that data point is assigned. We can now re-write the original objective function in Matrix notation as follows:

$$C = \mathbf{tr}[(\Phi - M)(\Phi - M)^T]$$

Next, using the fact $\Phi^t\Phi = K$, $\mathbf{tr}(AB) = \mathbf{tr}(BA)$, *and* $Z^TZ = L^{-1}$, we can rewrite this objective function as:

$$C = \mathbf{tr}[(\Phi - \Phi Z L Z^T)(\Phi - \Phi Z L Z^T)]$$

$$C = \mathbf{tr}[\Phi^T\Phi - 2\Phi^T\Phi Z L Z^T + Z L Z^T \Phi^T \Phi Z L Z^T]$$

$$C = \mathbf{tr}[K - 2KZLZ^T + ZLZ^TKZLZ^T]$$

$$C = \mathbf{tr}[K - L^{\frac{1}{2}}Z^TKZL^{\frac{1}{2}}]$$

Where K is the kernel matrix of similarity values and $L^{\frac{1}{2}}$ is defined as the square root of the diagonal entries. Instead of minimizing this objective function, we can set the problem up as follows:

$$\begin{gathered} maximize: \ \mathbf{tr}[L^{\frac{1}{2}} Z^T K Z L^{\frac{1}{2}} - K] \\ w.r.t \ to: Z \in 0, 1^{nxk} \\ subject \ to: Z1_k = 1_n \end{gathered}$$

This particular optimization problem is very difficult to solve due to Z being composed of binary decision variables, but we can relax this problem by setting $ZL^{\frac{1}{2}} = H$ and letting H take on arbitrary real values, while setting orthogonality constraints. This is possible because $Z^T Z = L^{-1} \Rightarrow L^{\frac{1}{2}} Z^T Z L^{\frac{1}{2}} = I \Rightarrow H^T H = I$. Now we can re-write the relaxed optimization problem using the orthonormal constraint:

$$\begin{gathered} maximize: \ \mathbf{tr}(H^T K H - K) \\ w.r.t: \ H \in \mathbb{R}^{nxk} \\ subject \ to: \ H^T H = I \end{gathered}$$

This problem can then be solved by applying Kernel PCA on the kernel matrix and setting H to the k eigenvectors corresponding to the k largest eigenvalues (in other words, the columns of H are eigenvectors of K. Now using the fact that H is an approximation of the assignment matrix Z, such that we cold threshold the values in the rows of H such that the largest value in the row becomes 1. However, a more practical approach is to normalize all rows of H to be on a unit sphere, so that each row is a k dimensional representation of H on the unit sphere. K-means clustering can be performed on this normalized matrix and because the representations lie on the unit sphere, the algorithm is less sensitive to initialization.

In the scenario where we have multiple data modalities or multiple views, we can reimagine the transformation into feature space, originally defined as $\Phi(x_i)$, by assigning non-negative weights that sum to 1 to the corresponding data sample from each modality. In this way, we replace $\Phi(x_i)$ with:

$$\Phi_\theta(x_i) = [\theta_1 \Phi_1(x_i)^T, \theta_2 \Phi_2(x_i)^T ..., \theta_m \Phi_m(x_i)^T]$$

The vector $\mathbf{\Theta}$, is composed of the weights that need to be optimized in this problem. From here we can define the optimized, combined kernel k_θ:

$$k_\theta(x_i, x_j) = \langle \Phi_\theta(x_i), \Phi_\theta(x_j) \rangle = \sum_{m=1}^{p} \langle \theta_m \Phi_m(x_i)^T, \theta_m \Phi_m(x_j) \rangle = \sum_{m=1}^{p} \theta_m^2 k_m(x_i, x_j)$$

Once again, this is a convex combination of kernel matrices, which will result in a symmetric, positive semi-definite kernel matrix. This combination can be used to adapt the original kernel K-means trace maximization problem to accommodate the scenario with mutliple kernels:

$$maximize: \ \mathbf{tr}(H^T K_\theta H - K_\theta)$$

$$w.r.t: \ H \in \mathbb{R}^{nxk}, \ \theta \in \mathbb{R}^p_+$$

$$subject\ to: \ H^T H = I_k, \ \theta^T \mathbf{1}_p = 1$$

$$K_\theta = \sum_{m=1}^{p} \theta_m^2 K_m$$

This problem is solved by alternating between optimizing for H *given* θ and θ *given* H. The first step of this process is to randomly initialize the kernel weights θ.

2.3 Applications of Multiple Kernel learning Based Algorithms to Genomic data

MKL has been used extensively for the analysis and integration of genomic data and was in fact studied by the authors (Lanckriet et. al., 2002) who developed the support vector machine based approach. In this paper, yeast protein function was predicted by fusing amino acid sequences, protein complex data, gene expression data and protein- protein interactions (demonstrating the usefulness of kernel functions in converting disparate data types to a common representation). This approach improved protein function prediction significantly compared to using any single data type alone.[13] This application relied on Lanckriett and Bach's original formulation which corresponds to an L_∞ regularized solution in kernel space and an L_1 regularized solution in the dual space. Because these modes of regularization correspond to a sparse solution[14], this method can be thought of as one to select a small number of particularly important kernels (corresponding to different data types). While this may be informative in identifying the most relevant data sources in a problem, the model can be undesirable because it does not incorporate information from all data sources.[15] Because of this inefficiency in using all available data sources, an L_2 regularized solution was developed by Yu et.al. to generate kernel coefficients which were more evenly distributed amongst all available kernels.[16] The authors showed that this approach yielded better results over the sparse solution for gene prioritization tasks.

Beyond tasks which mainly have to do with predicting the relationship between specific genes and disease state, many applications of MKL seek to define the relationship between the broader characteristics of a disease and effective treatments for that disease.[17] In a previous study, Wang et.al use MKL methods to merge semantic descriptions of diseases, different types of descriptors for chemical compounds, and finally use a learned combined kernel as the input for a Support Vector Machine Model which subsequently is able to distinguish effective drug-disease pairs, from

[13] Kernel Based Data Fusion and its Application to Yeast Protein Function Prediction

[14] Pattern Recognition and Machine Learning - Chris Bishop

[15] Kernel Based Data Fusion Approaches for Machine Learning- Methods and Applications in Bioinformatics and Text Mining

[16] $L_2 norm$ multiple kernel learning and its application to biomedical data fusion

[17] Drug Repositioning by Kernel-Based Integration of Molecular Structure, Molecular Activity, and Phenotype Data

non-effective ones (i.e. predicting which drug would be efficacious against a particular disease). In a more molecular take on this application, the same authors used an MKL framework to predict interaction between drugs and potential targets.[18] In a similar manner, the authors combine chemical structure, pharmacological, therapeutic, and genomic data to provide disparate views on the pairwise similarity between different drugs. The researchers then used this combined kernel to train another SVM model to distinguish between known drug target interactions and a sampling of drug-target pairs which are not known to interact with each other. Predictions were improved using this data fusion approach, and moreover, data fusion allowed for the identification of novel drug targets of some chemical entities, providing a justification for the incorporation of side information during the prediction stage.

While the main applications of MKL approaches merge representations which correspond to different data modalities, there also exist some approaches where different representations can correspond to different notions of similarity of the same data object. By merging these representations through MKL, we can therefore "learn" a more complex similarity measure through the combination of hundreds or even thousands of simpler kernels. These different similarity representations can also be thought of as "weak" kernels. The optimal combination of these weak models then give rise to a stronger model, similar to an ensemble method such as random forest. In fact, some approaches in learning an MKL model utilize boosting and bagging to generate a final model.[19][20] In general this represents an interesting approach because it overcomes a potential downfall of kernel based methods; defining a similarity measure a priori and validating competing similarity measures via model performance after training. In this way these models are able to learn an appropriate similarity measure with minimal guidance from humans. Learning these similarity measures is typically based on the concept of kernel alignment which at a high level measures the similarity between two kernel matrices, one based on features of a problem and the other based on the target values of a problem.[21] Moreover this approach can be combined with boosting and bagging approaches as mentioned before.[22]

Learning kernel representations of a data object through MKL has been applied to the analysis of single cell sequencing data with the goal of learning an appropriate similarity metric which can deal with the high levels of noise and dropout events which are generally associated with single cell sequencing data.[23] Moreover, these researchers were able to utilize the learned similarity metric as an input to t-SNE[24], allowing them to visualize groupings of single cells through the projection of the similarity matrix into a lower dimensional subspace. In this way, the approach can also be utilized/interpreted as a dimensionality reduction approach, as the original feature space is not used for downstream applications in the processing pipeline.

In addition to these two distinct approaches (using MKL to combine different data sources vs. using it to combine many weak kernels to learn a complex feature space

[18] Kernel-based data fusion improves the drug–protein interaction prediction

[19] Easy multiple kernel learning - https://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2014-129.pdf

[20] Gonen Review – Multiple Kernel Learning Algorithms

[21] On Kernel-Target Alignment

[22] Linear Combination Using Boosting

[23] SIMLR: a tool for large-scale single-cell analysis by multi-kernel learning– https://arxiv.org/pdf/1703.07844.pdf

[24] Visualizing Data Using tSNE

for a single data source), there also exist approaches that lie in between these two applications. In the context of predicting disease progression, an MKL based framework with a sparse solution was used for the selection of both different data types and also for specific features within that datatype. This was accomplished by training the algorithm on many kernels which represented different feature subsets of four different data modalities. Specifically, features for each modality were ranked by t-test, and N kernels were derived from the list of top N features. So for a list with 10 features, 10 kernels could be computed and would have been derived from lists with increasing dimensionality.[25] This approach is interesting in that provides an alternative way of learning the feature space and combining different modalities. Additionally, it could be easily modified to accommodate a feature bagging approach, although this would be computationally taxing.

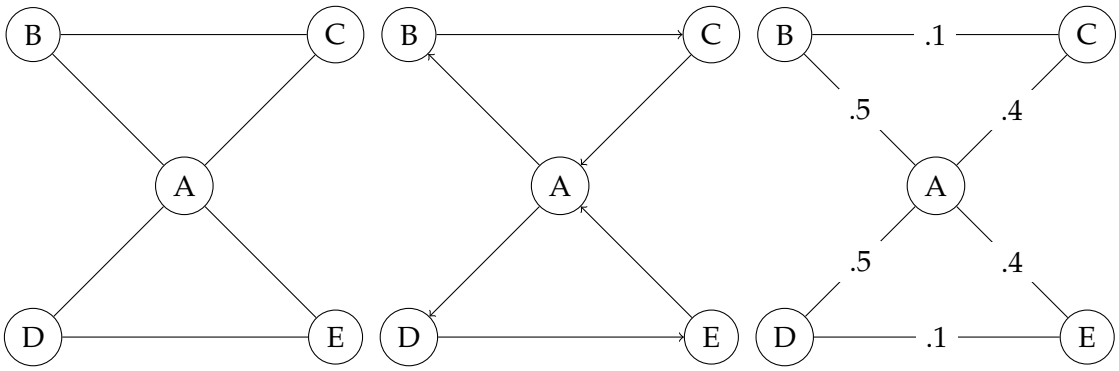
[25] A pathway-based data integration framework for prediction of disease progression

Chapter 3

Network Based Methods

3.1 Graphical models and graph integration

In a manner similar to kernel based methods, graphical models examine the relationships between samples to create networks of nodes representing individuals and connections called edges.[1] Connections that exist between nodes in a network which is meant to model similarity between samples are typically undirected as they only model class membership in many cases. Edges in a graph are encoded in an adjacency/ weight matrix which is symmetric (and not unlike the kernel matrix). In an unweighted graph, the cell of an adjacency matrix will be encoded as 1 if there is edge between nodes i and j of the graph, or 0 if no such connection is present. For weighted connections, edges can take on continuous values and are determined by some similarity metric. Existence of a weighted edge might be determined by some threshold value after the pairwise similarity matrix has been calculated. There are many ways to determine the magnitude of weights between the edges depending on the notion of similarity that is used. Like kernel methods, graphical models are a way to put disparate data sources into a common representation.



(A) An undirected, unweighted graph (B) A directed, unweighted graph (C) An undirected, weighted graph

FIGURE 3.1: Simple Versions of Graphical Models: Panel A shows the simplest graph structure, where binary relationships between nodes are illustrated by the presence or absence of edges. Modifications to these models include adding directionality to the relationship (B) or assigning a strength to the relationship (C)

[1] Koller Graphical Models

In the case where different data sources are being represented by graphical models, the resulting networks can be compared and/or combined. There exist a variety of ways to do this. One of most simple ways would be to create a graph with the same group of nodes and to take the intersection of edges between all graphs – that is, consider the edges that all graphs have in common and disregard any edges that have disagreement. A more relaxed of this approach would yield a graph which was described by the union of all the edges. Note that these approaches would only be appropriate for combining unweighted edges (or by discretizing weighted edges to binary values[2]). More advanced methods of integrating multiple graphs take inspiration from the Multiple Kernel Learning problem and formulate the corresponding problem as a quadratic programming problem using the weighted sum of the Laplacian matrices of each graph.[3] The MKL formulation is conveniently applied to the graph Laplacian as it is also symmetric and positive semi-definite.

3.2 Combination of Graph Laplacians through convex optimization

As previously discussed, individually derived graph based models can lend themselves to integration into a single graphical model by adopting a convex optimization framework which utilizes the graph laplacian much as it would use the kernel matrix in MKL approaches. To start, we will consider the process of a building a graphical model based on a single data source. Before we can achieve this, we must first discuss the defining structures of a graphical model; that is, the adjacency matrix and the laplacian.

Let us consider a graph $G = (V, E)$, where G is a graph which is represented by set of vertices (also called nodes) and edges (which represent connections between those nodes). An adjacency matrix (or weight matrix) defines the presence of all edges between nodes of a graph. If we let A be our adjacency matrix, then in the case of an unweighted, undirected graph, $A_{ij} \in 0, 1$, where the entry is equal to 1 if an edge exists between V_i and V_j. For graphs that have weighted edges, A_{ij} can take on any real value which reflects the weight of the edge and typically the strength of the relationship between nodes V_i and V_j. The laplacian is derived from A and a degree matrix D:

$$L = D - A$$

Where $D = \sum_i A_{ij}$ which represents the degree to which a node is connected with other nodes on the graph.

In a semi-supervised learning framework, a collection of n nodes ,V, are labeled such that all nodes can be combined into the same vector regardless of whether their identity is known or not. For the first p nodes in the label vector $V_i \in -1, 1$, for $i = 1, ..., p$. The next $n - p$ nodes are not labeled and thus $V_i = 0$, for $i = p, ..., n$. With this collection of labeled and unlabeled nodes, our task is to figure out the unknown labels using the underlying graph structure.

[2]Mayan Lab presentation–coursera

[3]Tsuda K, Shin H, Scholkopf B. Fast protein classification with multiple networks. Bioinformatics 2005;21(2):i59–65.

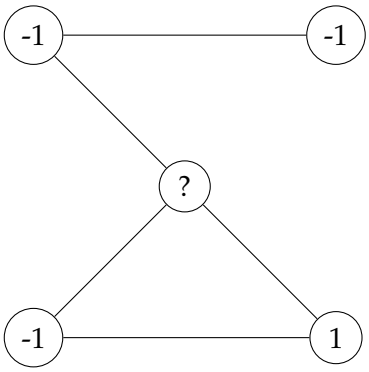


FIGURE 3.2: Graph structure of a semi-supervised learning problem: In this model, the graph is composed of a mixture of labeled and unlabeled nodes. Identities for unlabeled nodes are learned through its neighbors, and neighbors of a node are determined by distance or similarity measures between the nodes.

With this general structure defined, the approach then tries to define a score for each node, resulting in a score vector $\mathbf{f}$, where $\mathbf{f} = (f_1, f_2, ..., f_n)$. The goal is to find this vector f such that two criteria are met; the first being that the score f_i is close to the node's label y_i and the second being that the score a node should be similar to the scores of the nodes that surround it. Using these two criteria, we can attempt to minimize the following objective function:

$$minimize \sum_{i=1}^{p} (f_i - y_i)^2 + \mu \sum_{i=p+1}^{n} f^2 + c \sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$$

In this objective function, the first term corresponds to the squared error between a node's score and its label. The second term seeks to control the scores of unlabeled nodes so that their scores may be reasonable. Finally the third term is present to control the tradeoff between smoothness and the function loss. The contribution of this term is controlled by the parameter c. Using the case where $\mu = 1$, we can re-write the objective function as such:

$$minimize(\mathbf{f}) : \ (\mathbf{f} - \mathbf{y})^T(\mathbf{f} - \mathbf{y}) + c\mathbf{f}^T L\mathbf{f}^T$$

The solution for this optimization problem can be solved for analytically:

$$\mathbf{f} = (I - cL)^{-1}\mathbf{y}$$

Given this objective and solution for a single network, we can exploit this general structure to accommodate multiple networks. To accomplish this we must first re-write the objective that will allow us to introduce multiple laplacian matrices:

$$minimize(\mathbf{f}, \gamma)(\mathbf{f} - \mathbf{y})^T(\mathbf{f} - \mathbf{y}) + c\gamma$$

$$subject\ to : \mathbf{f}^T L\mathbf{f} \leq \gamma$$

For the scenario where we have constructed m networks corresponding to different data modalities or different constructions of the network, then we can utilize the above formulation like so:

$$minimize(\mathbf{f}, \gamma)(\mathbf{f}-\mathbf{y})^T(\mathbf{f}-\mathbf{y}) + c\gamma$$

$$subject\ to : \mathbf{f}^T L_k \mathbf{f} \leq \gamma,\ k = 1, ..., m$$

We can also express this problem in its equivalent dual form using lagrange multipliers:

$$maximize(\alpha, \nu)\ minimize(\mathbf{f}, \gamma) : (\mathbf{f}-\mathbf{y})^T(\mathbf{f}-\mathbf{y}) + c\gamma + \sum_{k=1}^{M} \alpha_k(\mathbf{f}^T L_k \mathbf{f} - \gamma) - \nu\gamma$$

$$subject\ to : \alpha_k, \nu \geq 0$$

Furthermore we can express this dual formulation in terms of strictly the dual variables α and ν by first solving for the inner minimization problem by setting the derivative with respect to *gamma* equal to 0. This yields the following equation:

$$c - \sum_{k=1}^{m} \alpha_k = \nu$$

By substituting this result into the previous equation, we obtain:

$$maximize(\alpha, \nu)(\mathbf{f}-\mathbf{y})^T(\mathbf{f}-\mathbf{y}) + \sum_{k=1}^{M} \alpha_k(\mathbf{f}^T L_k \mathbf{f})$$

To solve this equation, we solve for the derivative with respect to $\mathbf{f}$ equal to 0 and obtain the following:

$$(I + \sum_{k=1}^{m} \alpha_k L_k)\mathbf{f} = \mathbf{y}$$

Which is equivalently expressed as:

$$\mathbf{f} = (I + \sum_{k=1}^{m} \alpha_k L_k)^{-}1\mathbf{y}$$

In this formulation α acts as the weight vector for each individual graph. Furthermore since we found earlier that $c - \sum_{k=1}^{m} \alpha_k = \nu$, then we can see how the parameter c acts as a constraint on the size of α

3.3 Biomedical Applications of Graph Based Data Fusion

Graph integration approaches which leverage an MKL framework have been applied to patient diagnosis involving the integration of graphs which have been constructed in a semi-supervised manner as described in the previous section.[4] Briefly, labeled and unlabeled patient sample similarities were computed using a gaussian kernel and a graph was constructed using the rule that an edge would exist between nodes i and j if they were within the k-nearest neighbors of each other. These similarities were computed based on copy number variation (CNV), Methylation, Gene Expression (RNA), and miRNA. Following this construction, nodes which were unlabeled are then given labels by taking into consideration the labels of their neighbors. Graphs for different data modalities are then integrated using a convex combination of the Laplacian matrices for each graph explained above. This scheme was used to predict clinical outcomes for Glioblastoma Multiforme patients including short term/long term survival, tumor recurrence, and the grade of the tumor. Utilizing a graph integration approach consistently outperformed schemes which only used individual data sources. Furthermore, the authors of this paper were able to make conclusions about the usefulness of each data source and their respective impacts on the classifier's strength during prediction tasks. More specifically, when predicting initial vs. recurrent tumor classification in GBM, they found that the data types which offer direct insight into chromosomal rearrangement (CNV) were more heavily weighted in these tasks. The recurrence of tumors in these patients has been found to be due to chromosomal rearrangements, so it is interesting that the integration model reflects an increased importance on this data type. In a similar manner, the increased importance of gene expression data type when predicting short term and long term survival reflects the importance of protein expression dynamics in gauging how aggressive a cancer will be. The results of this study take into account the cumulative effect of changes in different levels of omic data. Since the cause of a disease can manifest itself at any point during the production of a protein, it is important to be able to take into account all available modalities which take part in that production process. Furthermore, the study showed the ability of data integration methods to successfully predict clinical endpoints for glioblastoma multiforme.

Another application of graph integration applied to biomedical problems is called Similarity Network Fusion (SNF).[5] This approach uses a diffusion based method based on message passing theory to optimize the combination of different data sources rather than combining graph laplacians. Briefly, the algorithm works by defining a row normalized weight matrix P and local affinitiy matrix S, whose entries are defined as:

$$S_{i,j} = \begin{cases} \frac{W_{i,j}}{2\sum_{k \in N_i} W_{i,k}} & \forall j \in N_i \\ 0 & otherwise \end{cases}$$

The local affinity matrix is therefore defined based on using a K nearest neighbors (KNN) based approach to determine which weights to maintain, while discarding those those fall outside a certain neighborhood. In the case where we have multiple data types, we define these matrices and iteratively update each normalized weight

[4] Synergistic Effects of Different Levels of Genomic Data for Cancer Clinical Trials
[5] Similarity network fusion for aggregating data types on a genomic scale.

matrix using the local affinity matrices derived from all available data types. This can be expressed mathematically as follows:

$$P_v = S_v(\frac{\sum_{k \neq v} P_k}{m-1})S_v^T,\ v = 1, 2, ..., m$$

Using this iterative update scheme, the matrix can then be learned and used for further clustering and classification. Additionally, by using local affinity matrices derived from all data types, the method ensures that weights that are present in all data sources are preserved in the final graph. Conversely, the iterative approach allows for the removal of low weight edges which are potentially not real and could only be contributing noise, affecting the performance of downstream learning algorithms. Low weight edges are preserved in the scenario where they are present in networks from multiple data modalities in the same local neighborhood. Additionally, since the local affinity matrices are sparse compared to the full weight matrix, the algorithm is computationally efficient.

The authors sought to apply this method to Glioblastoma Multiforme. This specific disease was chosen because of the fact that in previous studies, disagreements between the number of substypes of disease were dependent upon the data modality that been analyzed. These studies had found anywhere between two and four subgroups. When the authors applied Similarity Network Fusion to the problem, they found three distinct subtypes of the disease. Beyond this, they also found that using multiple data modalities allowed for the learning of a network which had more connections within the connected components of the network and less connections between the discovered subtypes. The subtypes that were discovered with this approach corresponded to subtypes that been previously described. Furthermore networks learned by the algorithm had the benefit in that that contribution of each data type to a particular edge was known and that the majority of edges were present as a result of all three data types analyzed in the glioblastoma multiforme dataset.

In this approach, networks are created for each data type by defining a similarity between samples for each modality and then constructing a graph representation where nodes represent samples (patients) and weighted edges represent similarities between samples. These can be visualized as either matrices or graphs. Combination of these networks occurs in an iterative fashion, distinct from the previously mentioned combination of Laplacians. Instead on each iteration, each network is updated to make it more similar to the other networks representing alternative data modalities. Convergence is reached when all networks have reached the same structure, representing a structural compromise between the originally derived similarity networks. The advantage of this approach being that low weight edges are eliminated in the final graph (unless they intermixed in a highly connected component of the graph), whereas high weight edges which might only be present in one graph are added to the final graph. In this way, the final representation of the patient network makes use of similarity information which made only be present in one specific data type.

Chapter 4

Unsupervised multiview learning experiments

4.1 Overview

The following chapters will describe a series of experiments in a range of unsupervised clustering problems that are designed to examine differences that exist between kernel based and graph based approaches. These experiments will be performed on both simulated and real world genomic data derived from patients with various types of glioma where either multimodal datasets are available or data integration methods can be applied to gain an alternative perspective to the typical methods of clustering and visualization. While we will be interested in assessing the "quality" of the discovered structure within the data, we will also look closely at features that define this structure and whether they make sense within the biological context. In the cases where clinical endpoints are available, we will see if there is a significance with respect to these clinical endpoints when considering the discovered structure by the multimodal learning algorithms.

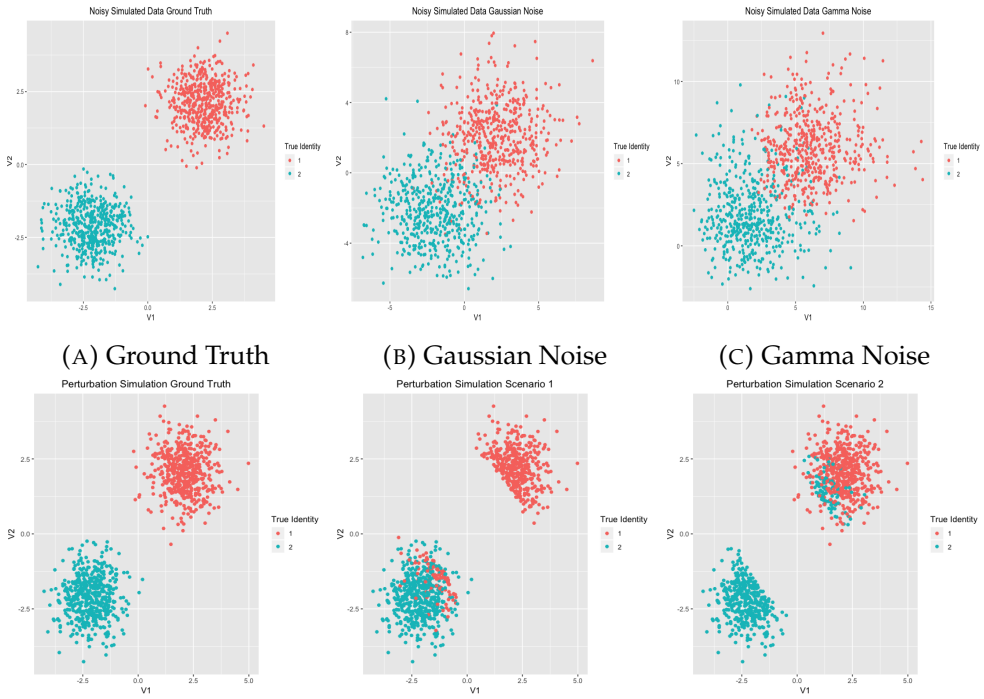
Broadly, our experimental approach will look like the following, although not all aspects of the following outline will be present in all of the experiments:

- Data Normalization and Preprocessing
- Data Transformation (i.e. casting data into a kernel or graph representation)
- Apply multimodal learning algorithm
- Analyze cluster structure using cluster quality metrics
- Visualization of clustering results
- Discriminative feature discovery and analysis of clinical traits between discovered groups

4.2 Unsupervised learning on simulated data

In order to assess the ability of multimodal learning techniques on a controlled dataset where we know the nature of the data (in terms of how well it describes different classes and what types of errors it might produce), we performed experiments comparing the performance of different unsupervised multimodal learning techniques on simulated datasets.

Simulated datatsets were generated according to the methods used in [1]. The first data set contains two linearly separable gaussian clusters in the ground truth state. This ground truth is then used to create two alternative views using Gaussian and Gamma Noise. The scenario tests the ability of a multiview algorithm to deal with the different types of noise that may be present unique to each data modality. The second simulated dataset, features the same two guassian clusters as ground truth. Each subsequent view is constructed such that a subset of points on the boundary of one cluster have been shifted to overlap with its neighboring cluster. This is then repeated for the opposite scenario (switching which cluster gets the perturbation) for the second view. Having no other information other than structure, a clustering algorithm trained on only one of these views would most likely have a difficult time capturing the overlapping populations, whereas taking both views under consideration would complementary information that could allow an algorithm to prevent that mistake.



(A) Ground Truth (B) Gaussian Noise (C) Gamma Noise

(D) Ground Truth (E) Misclassification 1 (F) Misclassification 2

FIGURE 4.1: Simulated multiview dataset: In the top row, a simulated multiview data set constructed by perturbing a ground truth (A) with Gaussian Noise (B) and Gamma Noise (C). The same ground truth (D) is used to created multiview dataset based on misclassification based on moving points from one cluster to overlap with the boundary of the adjacent cluster (E) and vice versa (F).

We used these two simulated datasets to examine the performance of similarity network fusion (SNF; a graph based method) and Multiple Kernel K-Means clustering (MKKM; a kernel based approach). A comparison of graph and kernel based method performance, along with the performance by the single view versions of these algorithms will be examined and discussed.

[1] Similarity Network Fusion

4.2.1 Preprocessing and data normalization

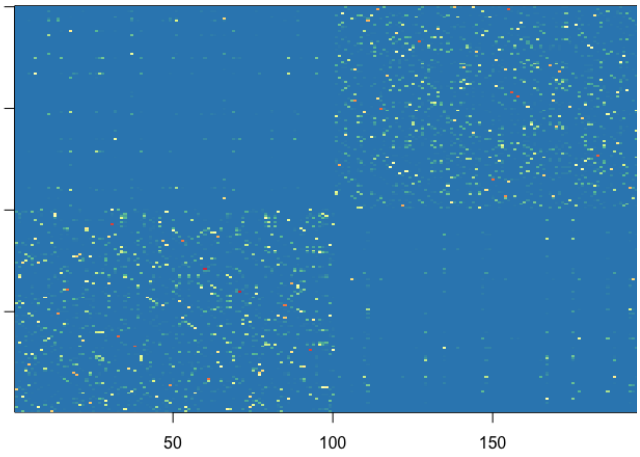
In the simulated experiments, preprocessing of the data is relatively straightforward for both SNF and MKKM. Centering and scaling of each feature in both data sets was performed such that each feature was centered at 0, with a standard deviation of 1. For SNF, euclidean distances were calculated between each of the data points in all views. Affinity matrices were then constructed from these distances. Instead of using a euclidean distance metric, an RBF kernel was used to calculate similarities between points. The width parameter (σ), was automatically selected by assuming that the observed distribution of simlarity values from the RBF kernel should be a normal distribution. Therefore, σ was chosen by optimizing the kolmogorov smirnov statistic comparing the observed distribution of similarities calculated by the gaussian kernel and a normal distribution centered at 0.5 a width of 0.18.

4.2.2 Making and Evaluating Predictions on simulated data

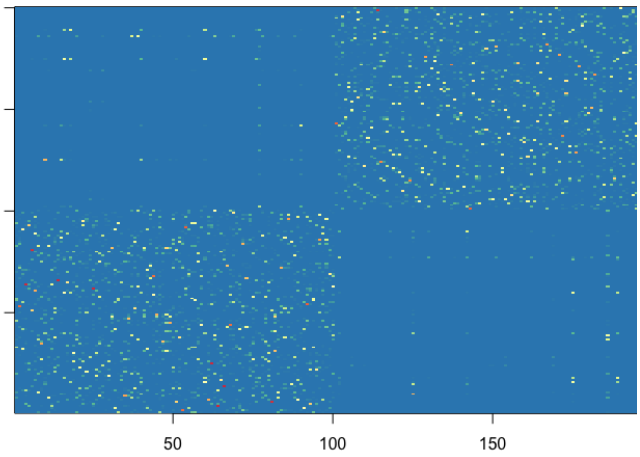
While we are doing unsupervised learning in most of our experiments in this study, we will typically not have label information to refer to as the ground truth. However, because our simulated data has a ground truth associated with it, we can used normalized mutual information (NMI) to assess the quality of the resulting groupings. Generally, mutual information $I(X;Y)$ quantifies the amount of information that is shared between two discrete random variables. The information between two discrete random variables is defined as:

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) log(\frac{p(x,y)}{p(x)p(y)})$$

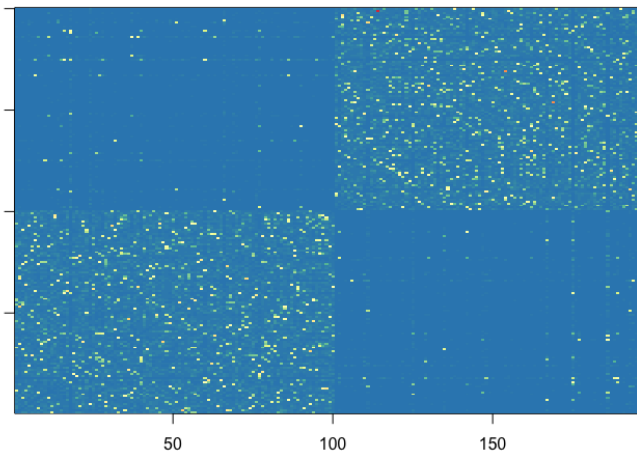
In its raw form, this value ranges from $[0, \infty]$, but we used a normalized version, which falls in the range $[0, 1]$. A lower value of the NMI between the two discrete random variables indicates complete independence between the two variables, and values closer to 1 indicate a strong relationship. The NMI between a predicted cluster assignment and a true cluster assignment vector can therefore be used to assess the quality of predictions from any clustering algorithm.



(A) Gaussian Noise



(B) Gamma Noise

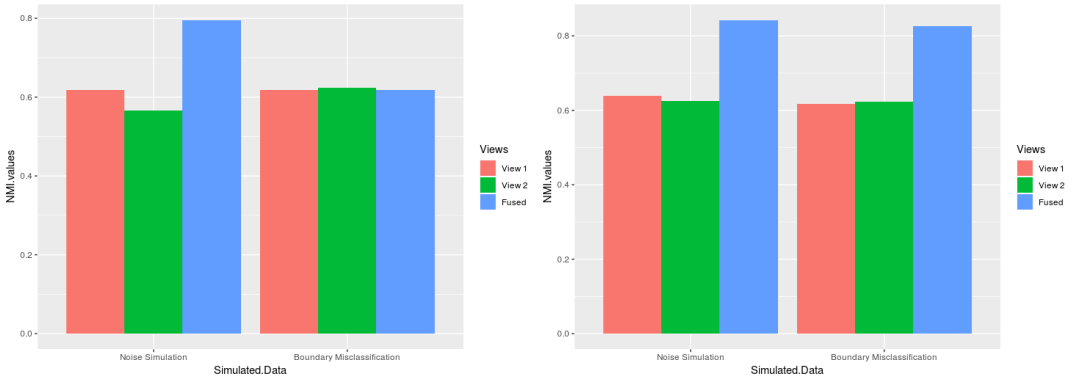


(C) Fused

FIGURE 4.2: Simulated data kernel heat maps: Heat maps of the kernels resulting from transformation of the Gaussian Noise (A) and Gamma Noise (B) simulations and the data dependent fusion of these two kernels (C)

In figure 4.2 we can see the NxN affinity matrices used in the SNF training process and the resulting fused affinity matrix visualized as heatmaps. All of the entries

on the heatmap are grouped by the cluster that they belong to (using the ground truth). Using this way to visualize, we see (and expect to see) a block diagonal pattern emerging depending on the quality of the grouping and also the quality of the similarity measure used. In both the Gaussian (A) and Gamma noise (B) panels corresponding to the similarities computed for each individual data view, we see that the general block structure is there, with there being some non-negligible similarity between samples that are not from the same group. In panel C, we see the result of similarity network fusion, the heatmap of the optimal combination of the affinity matrices from each individual data view. From the heatmap, we can see that the block diagonal structure of the fused view is much stronger than either of the Gaussian or Gamma noise views. We do however get the noise from both of the individual views in the fused view, however qualitatively speaking, it seems that this does not overcome the improved signal in the bloack diagonal structure.



(A) MKKM (B) SNF

FIGURE 4.3: Evaluating data fusion algorithm performance on simulated supervised data: Algorithm performance is displayed by plotting the Normalized Mutual information for the MKKM (A) and SNF (B) methods. In each panel, bars are grouped by the simulation experiment, with the fused result always being the last bar in the group.

In figure 4.3, we can see that the apparent improvement in the similarity metric by fusion of different data views also translates to improvement in clustering performance as measured by NMI. For the SNF method, fusion of individual data sources results in improved performance in both the noise and misclassification scenarios. For MKKM, fusion of the kernels from individual views only provides significant improvement predictive power in the noise simulation. The inability of MKKM to provide significant advantages in the misclassification simulation is perhaps indicative of the power of graph based methods in being able to determine the appropriate relationships in scenarios where class boundaries are very close or slightly overlapping.

4.3 Unsupervised Learning on Glioblastoma Multiforme Data: Methodology

Following the experiments on simulated data, the chosen multimodal learning approaches where tested on gene expression, miRNA expression, and DNA methylation data from 215 Glioblastoma Multiforme (GBM) patients from TCGA[2]. Several steps of pre-processing were performed prior to learning the the similarity matrices (via either method) as described in Wang et. al[3]. First, patients and features that were missing over 20% of their values were filtered out and were not considered further. For missing values that were below this threshold, K-nearest neighbor imputation was performed to replace the missing values. Features from the filtered and imputed data matrix were then converted to standardized z scores via the following method:

$$z = \frac{f - E(f)}{Var(f)}$$

Following the standard normalization procedure, principal components analysis was performed to reduce the dimensionality of the dataset, and the first 15 principal components were chosen for further analysis and were selected by the eigengap heuristic. It was especially crucial to reduce the dimensionality of the problem for computational reasons, but also to make the measurement of similarity between samples more robust, as distance measures in high dimensions typically are not helpful in making distinctions between samples. Following this reduction, similarity values were calculated for each view. In the SNF pipeline, euclidean distance was used to infer similarity, whereas for the MKKM method, the RBF kernel was used to calculate similarity. Following the conversion of each data view into a similarity matrix/kernel, SNF and MKKM were performed respectively. Predictions were obtained from each algorithm and were assessed for their quality.

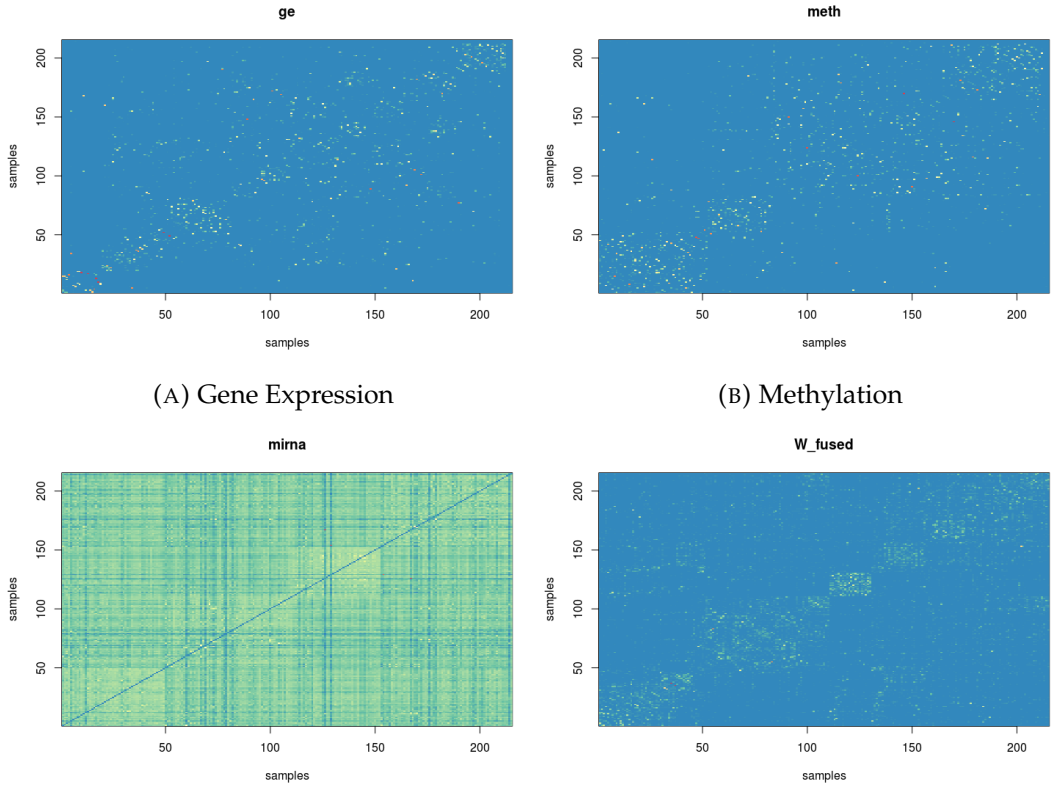
4.3.1 Evaluation of Unsupervised Learning Predictions

In real world scenarios, evaluating class assignments from unsupervised learning algorithms is notoriously difficult due to the obvious fact that there are no "ground truth" labels for which we can compare our predictions against. In these situations, not only are required to learn class assignments from the internal structure of the data, but we must also use this structure to determine the "quality" of the resulting clusters. This typically involves some sort of comparison between the distance to the assigned cluster center and the distance to the nearest neighbor cluster center. One such metric that makes use of this evaluation scheme is called the silhouette coefficient. The silhouette coefficient is defined based on two measures. The first measure A is the average distance between a sample and all other samples in the assigned class. The second measure B is the average distance between a sample and all other samples in the next nearest neighboring cluster. Based on these two values, the silhouette coefficient is then calculated as follows:

[2]https://cancergenome.nih.gov/
[3]Similarity Network Fusion

$$s = \frac{b - a}{max(a, b)}$$

This value s, falls between -1 and 1, with a higher score being indicative of a tighter (and hence higher quality) clustering. Silhouette scores that are around 0 indicate overlapping cluster assignments. For a set of samples, the mean silhouette scores over all samples are taken to be the silhouette coefficient.



(A) Gene Expression

(B) Methylation

(C) miRNA

(D) Fused View

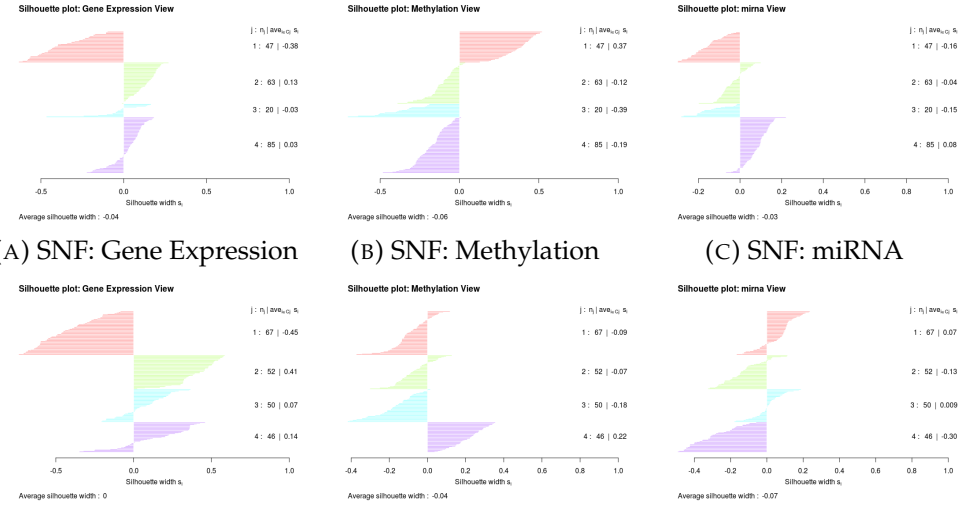
FIGURE 4.4: Kernel heat maps on GBM data: SNF affinity matrices constructed with RNA, methylation, miRNA, and fused view affinity matrices on data from patients with GBM. Rows and columns for all matrices are grouped by predictions made by the SNF algorithm

4.4 Unsupervised Learning on Glioblastoma Multiforme Data: Results

In figure 4.4, we can see another comparison of the similarity matrices learned from individual data views and the resulting fused similarity matrix. All of these data views are grouped by predictions made the SNF algorithm. We can see that across the individual data views, the similarity matrices possess varying degrees of block structure. The best of these, from a qualitative point of view seems to be the fused matrix which displays the strongest block structure out of all the similarity matrices

and also contains less similarity (noise) between samples that have been deemed to be in different groups.

We further assess the quality of the predictions made by SNF and compare them to those made by MKKM by taking a look at the silhouette plots in figure 4.5, constructed using the predictions made from each respective algorithm, and calculated on each individual data view. In the top row of the figure corresponding to silhouette values calculated using the group predictions made by SNF, we see that each different data view provides information for the formation of individual clusters. For example, the samples in cluster 1 recieve very high silhouette scores when calculating sample distances based on methylation, but not when calculating them based on gene expression or miRNA. For SNF, gene expression seems to be the most important in determing the structure of clusters 2 and 4, and miRNA also provides relevant information to cluster 4.



(A) SNF: Gene Expression (B) SNF: Methylation (C) SNF: miRNA

(D) MKKM: Gene Expression (E) MKKM: Methylation (F) MKKM: miRNA

FIGURE 4.5: Evaluating unsupervised learning on GBM data by silhouette scores: Silhouette scores were calculated by using distances defined by each individual data view using cluster membership predicted by SNF (top row) and MKKM (bottom row). Average silhouette scores for each cluster are reported in each individual panel

This can be contrasted to the silhouette values calculated using predictions from MKKM, where the quality of the MKKM predictions is mainly supported by the silhouette scores in the gene expresssion view, although the miRNA view seems support the membership of cluster 1, and the methylation data supports cluster 4. It is possible that while MKKM puts more emphasis on a specific view (a sparse view combination approach), SNF is able to utilize information from all three sources more evenly.

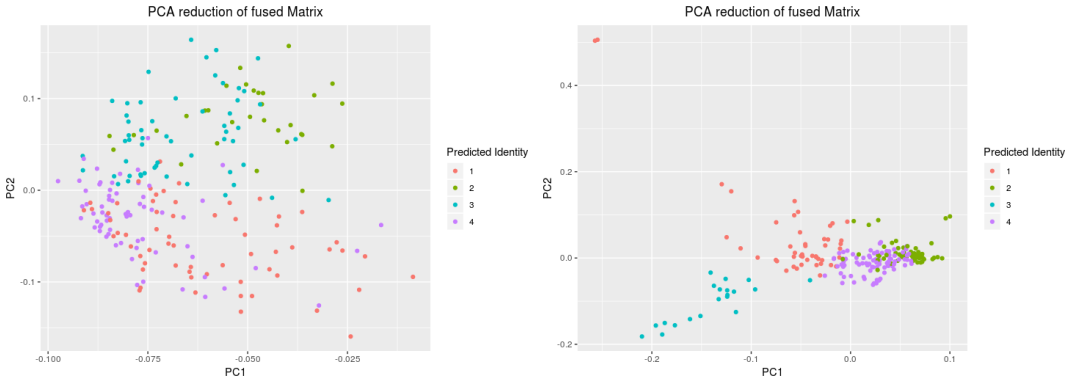
(A) Gene Expression (B) Methylation

(C) miRNA (D) Fused View

FIGURE 4.6: Comparing scatter plots derived from single views and the SNF fused views: The first two principal components of all individual data views and the SNF fused view were computed and used to make scatter plots of the data. Points in all panels were color coded by SNF predicted class membership.

To further examine the quality of the clusters in the single views versus a fused view, scatter plots were made to show the qualoty of the resulting clusters of all individual views and the resulting fused view (figure 4.6). The first two principal components were computed on each of the individual data views, and points were colored by predictions made by training the algorithm on predictions made from the single data view. In panel D, PCA was performed on the similarity matrix resulting from SNF, and color coded by the predictions made by fusing all the data views. Interestingly, we can see here that there is almost no separation in the scatter plots that have been derived from the individual data views, where as the scatter plot generated from from SNF yields compact, non-overlapping clusters. Obviously given this structure in a reduced form of the data it is not difficult to image that any learning algorithm would have an easier time learning groupings of samples from this representation of the data.

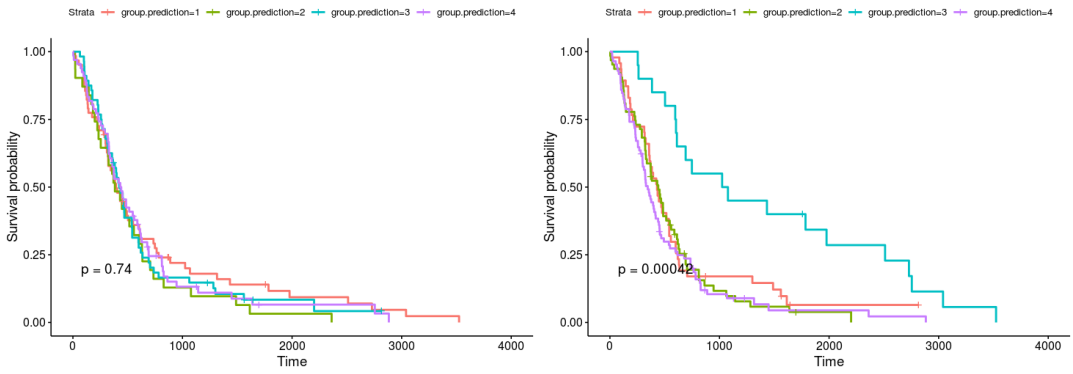
We also compare the results of SNF and MKKM by looking at the resulting scatter plots in figure 4.7. Both algorithms do a relatively good job of finding representation that are able to separate the predicted groups, howevere, it seems that the information from MKKM is more spread out over different principal components, where as the groups predicted from SNF could almost separated using only one dimension following PCA of the similarity matrix. This may be an important consideration when considering the ability of each algorithm to scale.



(A) MKKM dim embedding (B) SNF Dim Embedding

FIGURE 4.7: Comparing scatter plots derived from MKKM and SNF fused views: The first two principal components of the MKKM and SNF fused views were computed and used to make scatter plots of the data. Points in all panels were color coded by MKKM and SNF predicted class membership respectively.

Next, the clinical relevance of the groupings learned by each of the multiview methods was examined. In short, survival data for each patient was used to construct Kaplan-Meier curves to examine the ability of each algorithm to identify subgroups of patients that had more aggressive forms of the disease and thus had different survival times. Only SNF's predictions resulted in the identification of a patient population whose survival curve was significantly different from other predicted patient groups (Figure 4.8). The patients that were in the group experiencing longer survival times corresponded to the previously defined proneural subtype of GBM which typically coincides with IDH1 mutation and and PDGFRA amplification.



(A) MKKM dim embedding (B) SNF Dim Embedding

FIGURE 4.8: Survival curve comparison between MKKM and SNF predicted patient groups: Patient group predictions were used to construct patient survival curves for both MKKM (A) and SNF (B). Significance as computed by the Cox-Proportional Hazard ratio is reported in each panel.

In this section we have seen that using multiview learning algorithms can be useful in discovering sub-groups of patient populations that translate to clinically relevant distinguishing characteristics and can help researchers find markers for identifying particular subgroups of interest. We have also compared two distinct approaches

that can be used to accomplish this with respect to the quality of the predictions, learned structure, and the resulting discriminative molecular and clinical phenotypes arising from the resulting subgroups. In these particular experiments, it seems that graph based methods consistently outperformed kernel based methods. This seemed to be true in both low dimensional and high dimensional settings. One potential explanation for the superior performance of the graph based method in our context may have been due to its ability to take non-linear combination of the individual data views, whereas the kernel based method is restricted to linear combinations of the data.

While, we have shown that it is indeed possible to use multiview learning methods to discover subtypes of a population using multiple levels of disease data, it is also possible to use multiview methods to integrate datasets which represent the same modality, but have differences due to technical artifacts that are present due to collection of data. Furthermore, it also possible to utilize these multiview technologies to learn an appropriate similarity between the samples under study. This is particularly useful as it obviates the need for parameter tuning that is typically required in most learning settings. We will use these applications of data integration in a single cell sequencing context to identify oncogenic programs in patients with diffuse midline glioma (which is inclusive of diffuse intrinsic pontine glioma)

Chapter 5

Applications of multiview learning in Diffuse Intrinsic Pointine Glioma

As we have seen in previous chapters, multiview learning methods can be used to strengthen predictions by taking into consideration the union of multiple incomplete, but complementary view points and by providing robustness to noise in any of the data views. To state the obvious, these methods are therefore useful in scenarios where different data views communicate different notions of of similarity between samples and by themselves may each provide an incomplete notion of similarity between subjects (and the inferred divisions between groups of subjects based on similarity). Multiview methods have already been shown to perform well because of these advantages in general biomedical domains. We can further define the types of problems that would benefit the most from this approach by focusing on those diseases which have a strong genetic component and employ multiple underlying pathogenesis mechanisms to arrive at the same phenotype.

5.1 An Introduction to DIPG

Diffuse Intrinsic Pontine Glioma (DIPG) is a rare pediatric brain cancer that affects roughly 300 childeren each year, most often between the ages of 5 and 9.[1] Because of the young age at which the tumor manifests itself, the disease is thought to have a strong genetic component. The tumor starts in the brainstem, arising from glia which are a group of cells whose role is typically to support neurons. While DIPG is a rare disease overall, it happens to be more common than other types of pediatric brainstem tumors and accounts for around 80% of all cases. Unfortunately, on top of being the most common pediatric brainstem tumor, it is also the most aggressive. The poor prognosis of the disease is due to the diffuse embedding of the tumor within the brainstem tissue, posing a huge threat to the cranial nerve which controls many essential, life-sustaining functions. The median survival time from diagnosis is less than a year. Needless to say, this is a disease that is in desperate need of attention by the research community.

Even with more attention on this disease area, methods still need to be developed to more effectively research the disease. DIPG has been difficult to study due to its relatively low occurrence and lack of data surrounding the disease. Because of

[1] www.childrenshospital.org/conditions-and-treatments/conditions/d/diffuse-pontine-glioma

the tumor's diffuse nature in the brainstem and the mainstream adoption of MRI based screening methods for diagnosis of the disease, biopsies of tumor tissue are not routine and samples are typically post-mortem. Issues with these samples are typically due to tissue quality and they represent a state of the tumor right at the patient's rather than earlier state, where the patient was more treatable.[2].

5.2 New Research Tools for DIPG

While the situation is indeed dire for DIPG, technologies are improving rapidly to profile tumors with unprecedented resolution and the ability make sense of high resolution datasets. Recently, technologies have become available on a consumer scale to profile tumors at the single cell level[3]. Most of these platforms are also able to measure multiple modalities (genomic, transcriptomic, methylomic) at the single cell level. In most cases data are generated from a single modality, but multi-modal data at the single cell level are becoming available.[4]

Multiview learning is important, not only in contexts where we wish to unify information across data modalities, but also in situations where samples are rare and where we need to be extra efficient about information extraction. For instance, in biomedical and drug discovery contexts, where sample size of experiments are small scale relative to other application areas. Given that DIPG is a rare disease that is hard to treat and hard to even study, it makes sense to use a multiview learning approach to maximize the information extracted from each and every case that is encountered.

5.2.1 Recent Progress towards Understanding the Molecular Basis for DIPG

DIPG, like most cancers, displays considerable intratumor and intertumor variability at the molecular level. The main overall hypothesis for the appearance of DIPG in an individual revolves around development, since the disease manifests itself at a crucial developmental stage in the patient's life. Previous approaches have looked for neural precursor cell types which appear to have increased activity during the ages where DIPG is most common. Monje et.al. (2011) observed a cell type which displays typical markers of neural progenitor cells (Nestin and Vimentin), along with a characteristic marker for terminally differentiated oligodendrocytes (OLIG2). Interestingly, these cells have activity in the Hedgehog Pathway which regulates many developmental and oncogenic programs, and modifying the hedgehog pathway prevented DIPG derived neurospheres from continued growth. [5]. Other studies have found that RNA expression serves as a useful data source for distinguishing DIPG for tumors that manifest themselves in other regions of the developing brain. Using gene expression data, it was found that brainstem derived tumors are easily distinguishable from supratentorial tumors, while being very closely related to midline and thalamic tumors, suggesting a closely related origin. Moreover, in the referenced study, it was found that gene expression can be used to classify DIPG into 2 subtypes. The first of these identified subtypes exhibits mesenchymal and proangiogenic characteristics as well as various stem cell markers, indicating the ability for

[2]Diffuse Intrinsic Pontine Glioma: Poised For Progress
[3]https://www.nature.com/articles/nmeth.2771
[4]https://www.cell.com/trends/genetics/fulltext/S0168-9525(16)30169-X?code=cell-site
[5]Hedgehog-responsive candidate cell of origin for diffuse intrinsic pontine glioma

the tumor to renew itself. A second oncogenic program within the sampled DIPG tumors was associated with the strong overexpression of the gene PDGFRA, resulting in the dominating presence of cells with oligodendritic features. Of these two the subtypes, the appearance of an oligodendroglial phenotype in cells that composed the tumor resulted in a worse outcome for the patient (i.e. the second tumor type had a median survival time that was 5 months less that then first subtype).[6]

In perhaps one of the biggest breakthroughs in terms of understanding DIPG pathogenesis, Wu et. al. (2012) demonstrated that mutuations at the genomic level of gene H3F3A that resulted in the H3K27M substitution were present in 78% of DIPG patients. [7]. Mutations in this gene, which encodes for histones H3B and H3A, lead to the relaxation of a general repression mechanism for all DNA associated with the histones. Specifically, the H3K27M form of these histones, have been shown to inhibit the Polycomb Repressive Complex (PRC2) through binding of its catalytic subunit, EZH2. However, even though PRC2 activity is decreased globally for H3K27M cells, some genes are able to retain methylation at H3K27, which turns out to be necessary for DIPG cells to survive. [8]. From this particular study, we can see that the authors needed to employ analyses of both methylation data, as well as transcriptomic data to arrive at a potential mechanism of pathogenesis for this disease. Furthermore, through analyzing both of these data views in tandem, the researchers were able to verify the translational value of a mouse model they established, the first to show significant homology between the mouse version of the disease and the human version of the disease. Translational models such as this are very important with respect to screening and advancing a more diverse set of compounds with potential therapeutic benefits for patients with the disease. This progress was enabled via multi-modal analysis.

While the genetic underpinnings of the disease and their consequences have been well characterized at the tissue level, an in depth understanding of the cellular composition of tumors was still needed. In a study by Filbin et.al (2018), single cell sequencing was performed on six samples from patients with Diffuse Midline Glioma (a recently recognized disease grouping that includes DIPG), that possessed the H3K27M mutation. By using a single cell approach, the authors were able to extract the maximum amount of information from each tumor the profile. The success of efficient information extraction was two fold in that the researchers were not only able to get the levels of various genes in single cells, but were also able to infer sequences of each transcript, allowing them to identify mutations in the sequences. With the this data indicating the level and mutational status of all genes at the single cell level, they used multi-view data to perform powerful analysis. They found that the majority of each tumor was composed of oligodendrocyte precursor cells. Differentiated malignant cells constituted a smaller fraction of each tumor. Additionally, because of the single cell approach that was used, the researchers were able to find four different oncogenic programs, demonstrating the diverse number of strategies that tumors employ to be viable. These included gene signatures related to cell cycle, astrocytic differentiation, oligodendrocytic differentation, and OPC (oligo-precursor cell) like programs. This study also confirmed the hypothesis that there is decreased

[6] Mesenchymal Transition and PDGFRA Amplification/Mutation Are Key Distinct Oncogenic Events in Pediatric Diffuse Intrinsic Pontine Gliomas

[7] Somatic Histone H3 Alterations in Paediatric Diffuse Intrinsic Pontine Gliomas and Non-Brainstem Glioblastomas

[8] EZH2 is a potential therapeutic target for H3K27M-mutant pediatric gliomas

PRC2 activity in H3K27m cells and that this repression may be necessary for differentiation of precursor cells into the terminally differentiated form. Furthermore, the team was able to show that diffuse midline gliomas are driven by a stem cell population that are similar to the precursors of oligodendrocytes, as evidenced through the expression of the OPC marker PDGFRA. These findings have provided additional insight to the cause of DIPG. These insights brought by this study were enabled by the incorporation of both transcript level and mutational status of each gene at the single cell level. As single cell multi-modal technologies become more mature, researchers can start to find answers to questions that were previously not feasible.

5.3 Using data fusion approaches on single cell data

Single cell RNAseq is immensely powerful because of the resolution that it provides with respect to revealing cellular subtypes within tissues. This ability has proven to be especially useful in development contexts such as development of a cell type from the stem cell to a terminally differentiated stage. While scRNAseq is very useful, the data does not come without its pitfalls. Similar to bulk RNAseq, the technology is subject to batch effects which results in differences that are not due to biology, but are technical in nature. This can be exacerbated when combining datasets that come from different platforms, different tissue donors, or different animals. There are also several other technical complications that can arise in scRNAseq, however we will not discuss them here. For a more detailed review of the technical pitfalls in scRNAseq, please refer to Hemberg et. al. (2018).[9]

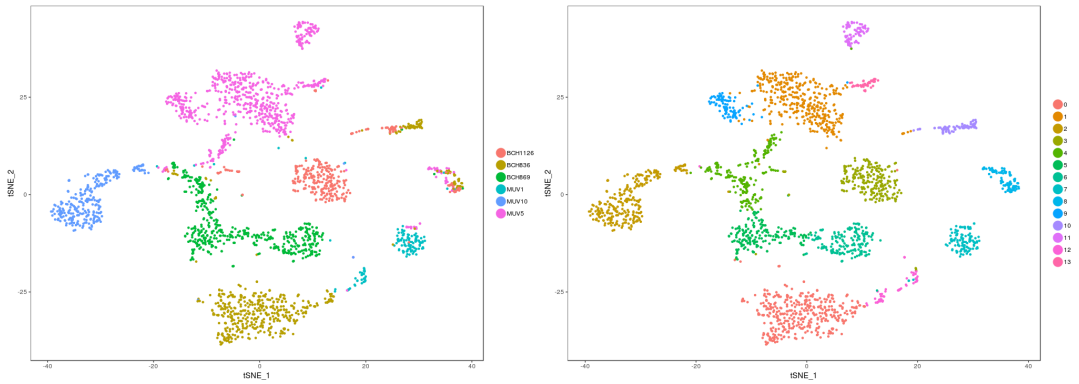
In this section we will discuss the use of two data integration methodologies, canonical correlation analysis (CCA) and multiple kernel learning (MKL) on single cell data derived from patients with diffuse midline glioma[10]. With these approaches, we will demonstrate how MKL can be used to learn a similarity from the data which can be subsequently be used for clustering or for visualization purposes. In addition to this, the application of CCA will be used to learn a subspace that is shared between samples derived from different patients, and to find latent components that summarize different oncogenic programs which are present in the total population.

5.3.1 Single Cell RNAseq data preprocessing, normalization, and subspace projection

We will briefly describe the process of preprocessing and normalizing single cell data, although more detail can be found at the Seurat website. Raw single cell data from the Illumina HiSeq 500 plaltform were processed using the Seurat R package. For quality control purposes, cells that had more than certain threshold of mitochondrial DNA were excluded from analysis since they are most likely low quality (dead) cells. Likewise, cells that contain too many or too little transcripts were excluded from the analysis as they were most likely dead or doublets. Following this, the raw transcript counts were log normalized and the data was scaled to account for different sequencing depths in the data by regressing the data based on the total number of transcripts detected in each cell. Following this, variable genes were detected by estimating a negative binomial distribution for each gene over all cells and

[9] https://hemberg-lab.github.io/scRNA.seq.course/index.html
[10] regev paper

taking those with the lowest dispersion coefficients. This is a feature selection step ensuring that we do not include static genes (and non-biologically meaningful in this context) genes in our further analysis. These variable genes were subsequently used to perform PCA to reduce to the dimensionality of the dataset. From the computed principal components, the first 25 were used for subsequent clustering and visualization steps.



(A) Batch (B) SNN clustering assignment

FIGURE 5.1: tSNE Emdeding of DIPG single cell data: Single cell RNAseq data from DIPG patients was reduced using principal components analysis and embedded into 2D space using tSNE. In panel A, cells are color coded by patient of origin and in B, cells are colored by predicted cluster membership.

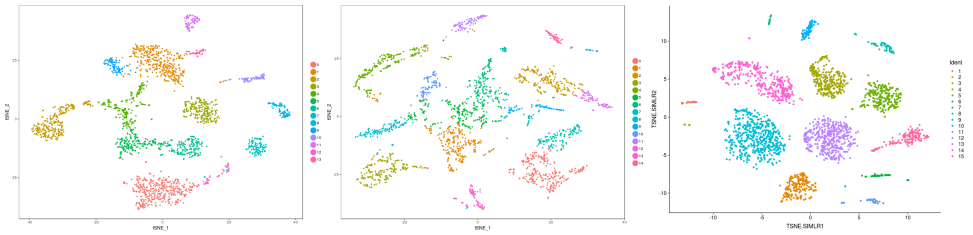
5.3.2 Clustering and embedding scRNAseq data

In figure 5.1, we can see the the embedding of single cell data in 2d space by applying tSNE to the principal components of the data. Clustering was also performed on the data in PCA space using Seurat's clustering algorithm based on Pe'er et al. and Xu et. al. [11][12]. In the first panel, the embedded cells are colored by the donor which they were derived from, while in the second panel, they are colored by the class prediction made after clustering. We can see that most of the cells from different patients tend to cluster separately indicating distinct cell identities between patients. There is a small amount of overlap between patients in a couple of clusters, suggesting that there may be some cellular populations that have common features between patients and are suggest to be malignant cells by Regev et.al[13]. In order to check that the separation between cells from different patients is due to biological difference and not to any technical artifact, we also embed the cells using CCA. The resulting embedding when color coded by batch still yielded a similar result, supporting the notion that there is truly a small amount of overlap between the tumors from these patients.

In figure 5.2, we can examine the effect of different embedding procedures on the resulting low-dimensional plot. Performing tSNE in the PCA and CCA spaces give somewhat similar results, with some of the CCA clusters having a more elongated nature than the PCA clusters. In the third panel, the tSNE embedding based on

[11] SNN cliq
[12] peer, facebook
[13] Dissecting H3K27M gliomas by single cell rna seq

SIMLR's learned kernel results in more discrete definition of clusters allowing us to more clearly define subpopulations of cells. Furthermore SIMLR's embedding is completely learned after designating the number of desired clusters, the only user-defined hyper-parameter. This is a significant advantage over the typical way of constructing these maps because both tSNE and SNN-cliq require many parameter choices. Inn practice it can be difficult to find the combination of parameters that yields a clustering/embedding result that makes sense when the resulting populations are examined in detail. For this other reason, using SIMLR to perform both clustering and embedding can be an attractive option to the standard and can also be used to provide an alternative perspective on the natural structure that exists in a single cell data set.



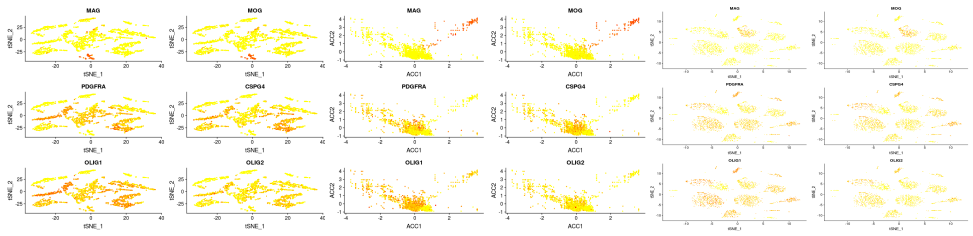
(A) TSNE embedding with SNN cluster assignments

(B) Aligned CCA embedding with SNN cluster assignments

(C) SIMLR embedding with KMeans cluster assignments

FIGURE 5.2: Effect of different embeddings on DIPG single cell layouts: DIPG patient cells embedded by tSNE on PCA, CCA, and SIMLR latent factors. CCA and PCA embeddings are colored by group predictions made by SNN-cliq and SIMLR embedding is colored by group predictions from SIMLR algorithm.

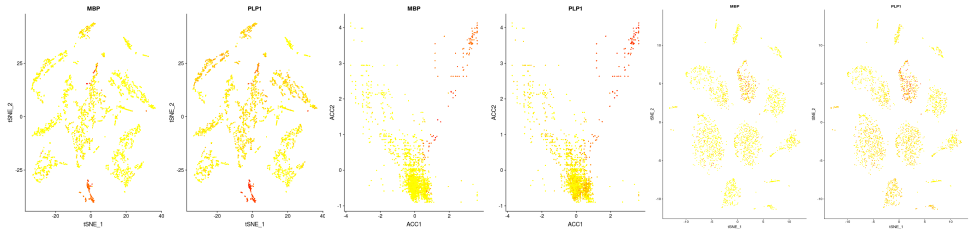
While examining embeddings and overlaying cluster identities can be informative, some of the most powerful information can be gained by looking at specific features that can be used as markers to identify discrete subpopulations of single cells. In Regev et. al., four distinct oncogenic programs are identified as being the most common routes leading to the pathogenesis of diffuse midline glioma. In this section we examine the expression patterns of genes associated with those oncogenic programs under different low dimensional embeddings. In figure 5.3, markers of oligodendrocytes are plotted in different low dimensional spaces and are color scaled to represent relative expression of each gene. In the first panel a tSNE based on the principal components of the data set is presented with the expression of six oligodendrocyte marker genes. While the expresion pattern of these genes is relatively concentrated, there are multiple clusters that express these genes in the embedding. If we contrast this to panel C which uses the SIMLR embedding to represent the cells in low dimensional space, we see that the expression of these markers is confined to one specific cluster. This is a more desirable representation of the cell space with respect to these markers because they represent one particular cell type and should thus be co-located.



(A) TSNE embedding (B) CCA Embedding (C) SIMLR embedding

FIGURE 5.3: Oligodendrocyte markers: Expression patterns of 2 Oligodendrocyte markers in PCA, CCA, SIMLR based tSNE embeddings.

In addition to finding markers of oligodendrocytes as being part of a distinct oncogenic program, markers of oligo-precursor cell differentation were also found to be a distinct program. Intuitively, this makes sense as a cycling stem-cell like population is thought to be essential to a malignant phenotype in many cancers. In the case of a glioma it should of course be derived from glial stem cells. In figure 5.4, we can see the expression patterns of genes which are associated with oligo-precursor cell differentiation. All of the different embeddings have relatively specific expression of these markers of differentiation, however the CCA and SIMLR embeddings are not optimal because they seem have grouped mature oligodendrocytes with OPCs, preventing us from picking up on the difference between these two cells without any prior information. In this case, the typical tSNE embedding is superior to the other approaches as it provides us with more information with respect to discriminating between different oligodendrocyte populations.

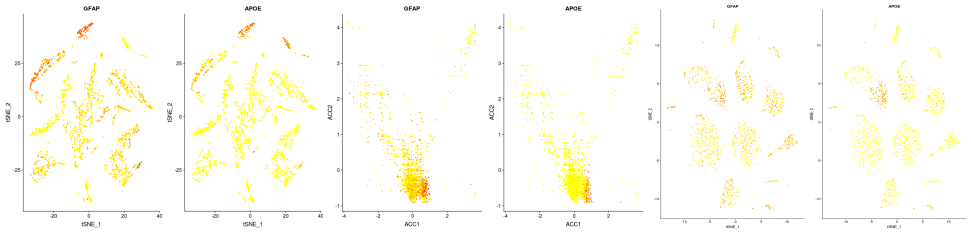


(A) TSNE embedding (B) CCA Embedding (C) SIMLR embedding

FIGURE 5.4: Markers of Oligodendrocyte differentiation: Expression patterns of 2 Oligodendrocyte differentiation markers in PCA, CCA, SIMLR based tSNE embeddings.

Figure 5.4 examines the expression of astrocytic differentiation markers. Looking at these markers, and taking the expression of our previous cell type markers into account, it is arguable that all three of these embeddings provide useful information with respect to cellular origin and distinction. In the tSNE embedding astrocytic differentiation markers are expressed in clusters that do not express any oligodendrocyte markers. This is useful information because these cells are in fact destined to become a different cell type than oligodendrocytes. However, it is worth noting that these cells which are in transition towards a final differentiated astrocytic state, still share a common origin with oligodendrocytes. So, depending on the question one is trying to answer, it may be more convenient or more "correct" to have an embedding which groups cells together which have a common origin, as is the case with these glial cell types. In the SIMLR embedding this is achieved. Furthermore, in the

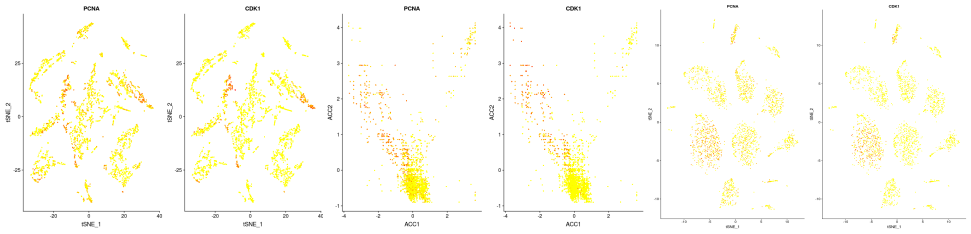
CCA embedding which is present in the same figure, these astrocytic differentiation markers are expressed in part of the embedding which is near cells which strictly express oligodendrocyte markers, without being distinct from each other. This particular embedding is more true to the nature of these in that the canonical correlates represent a dimension which is loosely related to cellular state, capturing both the common origin of these different cell states, while also giving us the transformation which would allow us to more easily train a discriminative classifier based on a minimal amount of information.



(A) TSNE embedding (B) CCA Embedding (C) SIMLR embedding

FIGURE 5.5: Markers of Astrocytic differentiation: Expression patterns of 2 Astrocytic differentiation markers in PCA, CCA, SIMLR based tSNE embeddings.

In addition to these cell markers pertaining to glial cell type signatures, we also examined the cell cycle markers which were reported to be found as a distinct oncogenic program in DIPG. The CCA embedding is particularly interesting to pay attention to for these markers as figure 5.6 illustrates. With the tSNE and SIMLR embeddings, these markers are coexpressed with cells that express CSPG4 and PDGFRA. In the CCA embedding however, these markers occupy their own specific space within the embedding, showing it as distinct program which is unique from malignant cells that are expressing glial cell or precursor cell markers.



(A) TSNE embedding (B) CCA Embedding (C) SIMLR embedding

FIGURE 5.6: Cell Cycle Markers: Expression patterns of 2 cell cycle markers in PCA, CCA, SIMLR based tSNE embeddings.

With these different embeddings and transformations we can see that it useful to utilize multiple approaches to represent cells and their gene expression in a low dimensional space. The ability to find a representation where one or two dimensions encode enough information to allow us to distinguish between these previously defined oncogenic programs shows that data integration methods are important even when we do not have different modalities to combine. This is because they allow us to see the data from another perspective, revealing patterns and relationships which may not have been apparent when examining the data using more traditional embedding methods. Furthermore, using relationships between data points to find low

dimensional representations of the data are important as we have already seen that distance metrics typically lose their meaning when applied to data in high dimensions.

Chapter 6

Conclusion

6.1 The Future of Multi-modal Analysis in DIPG Research

Through this thesis, we have made comparisons between both kernel and graph based multiview learning methods and have seen that both have their strengths. MKL frameworks have a strong theoretical base and offer a diverse set of modifications to the regularization term of the problem, resulting in distinct methods for kernel combination. These choices in regularization allow for the user to make choices between kernels that either weigh a particular data type more heavily through a sparse weight vector or alternatively allow for a more even combination of kernels. Graph based frameworks, on the other hand, offer a set of methods which are seem to be empirically robust in situations where we have few samples and a high number of dimensions. Both of these and potentially other multiview learning frameworks will have their applications in the years to come. In the biological domain, data fusion has already become increasingly important in the unified analysis of multi-modal datasets. It is often the case that a disease (especially a cancer) manifests itself via multiple mechanism and can find multiple strategies to survive in the face of both natural and man-made efforts to eradicate it (i.e. acquired resistance of cancers to therapeutics). Therefore it is essential to look at multilevel data to see the complete story of a disease. Analyses based on multi-modal single cell technologies will give us an unprecedented level of understanding about particular cell populations that we are interested in and will probably open up a number of new questions.

Beyond the approaches that were already explored in this thesis, more methods are being developed to tackle the problem of multi-modal data integration. In particular, deep learning based approaches are particularly promising because of their prior success in the image and natural language domains. [1] While these deep learning approaches may not necessarily be the best fit for omic data, it is reasonable to suggest that they will be useful in integrating information from image based data (MRI, images from phenotypic screens) and text based clinical data (doctor's notes, electronic health records). With a disease like DIPG it is important to tackle it from many different angles. Multimodal learning techniques give us a tool box which allows us to do exactly that.

[1] source