Adaptive Learning and Unsupervised Clustering of Immune Responses Using Microarray Random Sequence Peptides

by

Anna Malin

A Dissertation Presented in Partial Fulfillment
of the Requirement for the Degree
Doctor of Philosophy

Approved October 30, 2013 by the
Graduate Supervisory Committee:

Antonia Papandreou-Suppappola, Chair
Daniel Bliss
Chaitali Chakrabarti
Narayan Kovvali
Zoé Lacroix

ARIZONA STATE UNIVERSITY

December 2013

ABSTRACT

Immunosignaturing is a medical test for assessing the health status of a patient by applying microarrays of random sequence peptides to determine the patient's immune fingerprint by associating antibodies from a biological sample to immune responses. The immunosignature measurements can potentially provide pre-symptomatic diagnosis for infectious diseases or detection of biological threats. Currently, traditional bioinformatics tools, such as data mining classification algorithms, are used to process the large amount of peptide microarray data. However, these methods generally require training data and do not adapt to changing immune conditions or additional patient information.

This work proposes advanced processing techniques to improve the classification and identification of single and multiple underlying immune response states embedded in immunosignatures, making it possible to detect both known and previously unknown diseases or biothreat agents. Novel adaptive learning methodologies for unsupervised and semi-supervised clustering integrated with immunosignature feature extraction approaches are proposed. The techniques are based on extracting novel stochastic features from microarray binding intensities and use Dirichlet process Gaussian mixture models to adaptively cluster the immunosignatures in the feature space. This learning-while-clustering approach allows continuous discovery of antibody activity by adaptively detecting new disease states, with limited *a priori* disease or patient information. A beta process factor analysis model to determine underlying patient immune responses is also proposed to further improve the adaptive clustering performance by formatting new relationships between patients and antibody activity. In order to extend the clustering methods for diagnosing multiple states in a patient, the adaptive hierarchical Dirichlet process is integrated with modified beta process factor analysis latent feature modeling to identify relationships between patients and infectious agents. The use of Bayesian nonparametric adaptive learning techniques allows for further clustering if additional patient data is received. Significant improvements in feature identification and immune response clustering are demonstrated using samples from patients with different diseases.

ACKNOWLEDGEMENTS

While this dissertation is the result of years of focus and dedication, it would have been impossible to complete without the support and guidance of several individuals. I am immensely grateful for their constant encouragement, vigilance, and care.

I would first like to thank Dr. Antonia Papandreou-Suppappola for her constant support as my advisor and committee chair. This accomplishment would not have been possible without her wisdom and encouragement. My tenure with her as a grad student has been one of challenge, perseverance, and respect. Her sincere care and mentorship have greatly impacted the person I am today. I would also like to acknowledge Dr. Narayan Kovvali for his extremely helpful teachings, patience with my learning, and words of inspiration when I was intellectually stuck. His dedication to his work, colleagues, and students is greatly admirable.

Additionally, I am grateful to Dr. Chaitali Chakrabarti, Dr. Zoé Lacroix, and Dr. Daniel Bliss, for their diligence and thoughtfulness as members of my committee. I appreciate the time and support they have given as well as their demonstration of constant academic professionalism.

I would like to also thank my family for their never failing encouragement and support. My mother's influence has led me to be an independent professional, never doubting that I can do what I set my mind to. My father's constant challenge and drive has inspired me to ever greater achievement and capabilities. I am truly grateful for every moment that they have helped pushed me towards my goals, and encouraging me throughout these years. Both have played a huge role in my successful completion of this accomplishment.

And finally, I would like to thank Andrew who was with me during this work. I would like to thank him for believing in me when I did not, for pushing me on when I could not, and for making me take a break and relax when I swore I would not.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

Table Page

Table Page

LIST OF FIGURES

Figure Page

Chapter 1

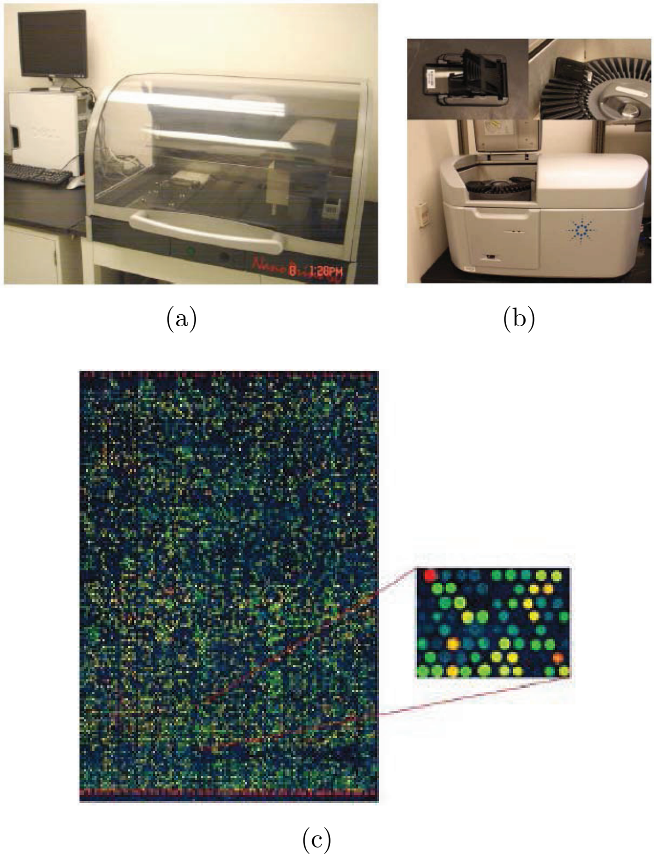
INTRODUCTION

The ability to detect disease pathogens by the use of peptide microarrays [1–3], which may be used to detect a variety of biological molecules [4–6], has led to the development of immunosignaturing. Immunosignaturing is a technique that has been devised to create a snapshot or fingerprint of patient pathology at a given point in time [7–14]. This is done by using microarrays preset with randomly generated peptides as a measurement device for patient samples with various antibodies. Antibodies will preferentially bind to peptide sequences based on the sequence order and three-dimensional shape via molecular associations specified by traditional organic chemistry interactions [15]. For peptides of sufficient length, multiple antibodies are able to bind to a particular peptide chain, as there are up to seven different epitope regions present [7]. As greater number of peptide chains, and hence a greater number of microarray spots, are included, the higher the resolution into a variety of diseases, and the more difficult data analysis becomes especially between multiple measurements. The focus of this work is to develop adaptive methodologies for immunosignaturing that are capable of discovering relationships between patient disease states and to group patients with similar disease states.

1.1 Motivation

One important aspect of immunosignaturing is the construction and interpretation of the microarray data. Randomly generated peptide sequences, in this case with a length of 17 amino acids and three additional linker peptides, are plated onto a glass slide capable of holding thousands of such plated samples [7, 9]. Each plated spot contains only one peptide type whose amino acid sequence and location on the plate are known [10]. When a patient sample is applied, molecules from the sample will preferentially bind to the plated peptides, producing a signature unique to the patient at a particulate point in time [11, 12]. An example of the microarray plate, printer, and slide reader is shown in Figure 1.1. This plate construction differs from traditional microarray work where pathology-specific, known, non-random sequences are plated instead of randomly generated [16]. The traditional methodology provides insight into a specific pathogen or gene rather than detecting a variety of pathological ailments [9]. While immunosignaturing is considered highly sensitive and inexpensive [17], some critical parameters of the immunosignaturing array include the number of amino acids included in the random chain, the number of overall sequences included in the microarray slide, and the time between pathogen exposure and microarray measurement [18]. As demonstrated in [9], immunosignaturing microarrays are capable of distinguishing disease presence at a variety of time points after exposure.

As more spots with unique sequences are added, data analysis becomes both challenging and repetitive. Additionally, there is a desire to increasingly add patient immunosignatures to account for population based variation [19, 20]. Statistical tools such as the t-test and analysis of variance (ANOVA) are often used for microarray analysis [21–23], and have been applied to immunosignaturing as well [24]. It should be noted that any down selection that occurs from these statistical tests would need to be fully repeated every time a new patient immunosignature microarray data set is received. Once peptides of interest are identified, further analysis or disease classification can be performed to investigate the binding regions [20, 25]. A variety of clustering and classification schemes were applied to different microarray data in order to aid in data analysis [26–29]. These methods were also used for immunosignaturing analysis [7, 10, 24], however, they required training data and are



(a) (b)

(c)

Figure 1.1: Peptide microarray processing and resulting data: (a) nanoprinter; (b) microarray reader; and (c) example slide and spots.

not adaptive. This is complicated further by the fact that each peptide can bind to multiple antibodies and each antibody can bind multiple peptides [30]. This does not easily allow for the addition of further microarray data for each patient, or for additional patients to be analyzed without re-implementing the algorithm. Additionally, these methods do not address the concern of considering fixed versus random data effects [31]. Supervised learning methods such as support vector machines (SVMs) were used for improved performance over unsupervised clustering methods [32]. The SVMs make use of distance measures in higher dimensional feature spaces for effective separation and classification. However, they require adequate training datasets, which may not always be available (e.g., in the presence of a new biothreat). Some model development for traditional microarray work was investigated to improve over SVM techniques [33], though a training set is still required. Furthermore, complicating effects for peptide arrays (such as multiple binding effects and variation between arrays) are examined in [34], which uses many of the techniques discussed thus far but applies them to screen out data not deemed useful. Of further interest is the investigation of underlying features within a factor model, as investigated by [35], and using principal component analysis (PCA) [9, 10]. However, the PCA method assumes that information content may be maximized across the same orthogonal bases which may not always be true. Further improvements to the PCA technique was performed using exploratory factor analysis [24]. Other latent factor models were investigated in [35] where gene interdependence was examined but individual known sequences were targeted. Additionally, factor modeling via beta process feature analysis (BPFA) was examined by [36] for human virus challenge studies paired with patient symptoms, but this only focused on determining the total number of factors and contributing genes in each virus challenge.

The presented classification methods and latent factor models are not adaptive to changes. Adaptive in this case can mean several things, such as extensible to additional incoming microarray data, model parameter adaptability on-the-fly for newly received data, and updated classification based on model factors. An adaptive clustering scheme based on the Chinese restaurant process (CRP) was presented in [37, 38] to cluster gene microarray data for gene relationships. A binary clustering model was presented in [39] that is based on latent feature analysis, but it assumes that the features act as latent individual clusters, which may be too restrictive. It is important that the immunosignaturing data model relies on flexible feature relationships that may not be immediately obvious or known to the user and may require user information. This approach is useful when those performing the analysis are not familiar with immunosignaturing, the patient population, or model fitting and classification algorithms.

Of further interest is the extensibility of the methods to multiple underlying states, including disease pathologies, for complex classification in disease diagnosis or for research cases that desire to show disease relationships between multiple diseases. It is restrictive to assume patients are afflicted with a single condition at any given time [40], especially since the presence of multiple diseases can require additional care considerations [41]. Furthermore, disease relationships may be of interest as diseases may be related leading to similar treatments [42]. Single disease classification for immunosignaturing has been investigated for a variety of diseases including Alzheimer's [10], influenza [9, 11], glioblastoma [20], and pancreatic disease (including type 2 diabetes, pancreatic cancer, pancreatitis, and pre-stage pancreatic cancer) [8]. No direct extension to multiple states is considered, and the adaptive methods in [43, 44] are not directly applicable as they assume single disease states. Additionally, these adaptive methods do not provide diagnosis information after classification, though it would be easy to extend these to classification by the incorporation of a few known immunosignatures for each known disease. Some similar adaptive methods for multi-state membership have been discussed in [45], but this method uses the Indian buffet process "IBP" to model each different category and relates each separate feature to each view. Building on the immunosignaturing results discussed in [43, 44], this would necessitate multiple BPFA runs to generate all the different categories, which is computationally intensive and redundant. Another method is discussed in [46], using the IBP to create multiple cluster membership relationships. However, this approach returns to the assumption that a single feature is indicative of a single state and that the clustering feature space is not the same as the latent feature space, which previous work has shown may not be the best approach for immunosignaturing. Improved classification was achieved using latent feature spaces for immunosignaturing [44]. As such, there is a desire for an adaptive method that provides diagnosis or disease relationship information that is capable of multi-disease analysis.

1.2 Thesis Contributions

There are two major contributions in this thesis work that propose unsupervised or semi-supervised adaptive learning clustering algorithms for immune responses. The first contribution concentrates on patients with immunosignatures resulting from a single state. The second contribution is on patients with multiple underlying states. Note that we published some of this work in [43, 44, 47, 48].

1.2.1 Single State Clustering

The first part of this work focuses on obtaining immunosignature feature models for immunosignaturing data analysis. These models mathematically represent relationships between patients and various single underlying states (i.e. diseases) with limited *a priori* patient or disease information. Using these features, we propose methods for adaptive clustering that allow for new incoming patient data, without requiring us to re-process previous patient or supplemental training data. The main contributions for the first part of this work are as follows:

- We propose two different approaches for feature extraction that improve immune state separation. The first approach obtains discriminatory immunosignature features by modeling the distribution of normalized binding intensities using the beta probability density function. This allows for multiple distribution shapes whose parameters can be optimally estimated to succinctly encode information about the distribution of the peptide binding intensities. This is useful when states may have similar antibody occurrences but in different quantities. The second approach is an extension of PCA to identify features that encode highly variable data. The extension to a higher truncation value than previous methods accounts for greater than 90% of the immunosignature data, and it is useful when different significant antibodies are present but with similar overall antibody distributions. These two methods are used to determine visible spaces as well as for overall dimensionality reduction techniques.

- We develop an algorithm to extract hidden features from the visible features. Combining both the visible and latent features, we propose two clustering algorithms: a heuristic algorithm based on the output from the beta process factor analysis (BPFA) algorithm, and an adaptive Dirichlet process Gaussian mixture modeling (DP-GMM) algorithm. Both clustering methods are based on the novel interpretation of a modified BPFA binary feature matrix to allow for combinations of features to describe single states. This modification allows for the novel introduction of reward or penalty criteria for various clustering conditions that are application specific. The user can then account for a wide range of cases, including cases with a low tolerance for error and cases that are more tolerant to allow for feature relationship discovery.

- We develop methodologies for unsupervised identification of disease features linked directly to unique patient groups, using only the patient immunosignature median intensity values and requiring no training data or additional information. This is a novel application in the immunosignaturing space.

These main contributions are based on several proposed methods. The PREDICT (PCA REsolution with DPGMM for Immunosignature Classification Testing)

B-PREDICT (Beta PDF REsolution with DPGMM for Immunosignature Classification Testing) methods use DP-GMMs to identify groupings given PCA and beta distribution parameters, respectively [43, 47]. These methods assume that each patient belongs to only one group. By using the blocked Gibbs Markov chain Monte Carlo (MCMC) algorithm, Gaussian distributions are constructed for each group. This can lead to misclassification and disease state misdiagnosis. The DP-GMM based methods allow adaptive identification of groupings between immunosignatures using only the microarray median intensity data and further allows clustering of novel groupings without the need for a training data set. While these methods show promising results, for some cases the features are not always clearly associated with a single group.

In the Z-PREDICT (Z matrix from PCA REsolution and Discovery for Immunosignature Classification Testing), Φ-PREDICT (Φ PCA REsolution and Discovery for Immunosignature Classification Testing) and ZB-PREDICT (Z matrix from Beta PDF REsolution and Discovery for Immunosignature Classification Testing), beta process factor analysis (BPFA) is used to identify latent features within the data set and with the possibility of multiple feature grouping for each disease state [44]. Note that this happens after the visible features are identified and encoded in the data. In fact, this method produces a ”feature fingerprint” for each of the states which can then be used for clustering and ultimately disease identification. The BPFA is also able to highlight the relationship between the peptide sequences and latent feature groups, though further investigation of this phenomena is outside the scope of this work. This provides a method for identifying spots which may be related to particular diseases without re-running the microarray for each disease state separately. Again, by using a blocked Gibbs sampler, the posterior distribution parameters that mathematically describe these relationships can be estimated. These methods are further extended to include DP-GMM clustering in BIO-PREDICT (BPFA Including prOcessing with PCA REsolution and DP-GMM for Immunosignature Classification Testing) and BIOB-PREDICT (BPFA Including prOcessing with Beta PDF REsolution with DP-GMM for Immunosignature Classification Testing), leading to an adaptive clustering method that is paired with the adaptive model determination of the latent features.

The ability to adaptively determine underlying features for immunosignatures with the possibility of recognizing novel features over time is new to immunosignaturing. In both the DP-GMM and BPFA, new patient data may be considered as it is received without changing the clustering of previously analyzed patients. While we achieved good performance, BPFA clustering alone becomes difficult for immunosignaturing due to the multiple feature combinations and multiple patients. As such, adaptive clustering is shown to perform better when DP-GMM is combined with BPFA.

1.2.2 Multiple State Clustering

The contributions for the first part of this work focuses on single underlying states in patients, which may be too restrictive in practice. As a patient may have multiple disease states, each data entry needs to be classified into multiple groups. In addition to multiple diseases per patient, it is also possible that a single disease has multiple stages or state relationships need to be explored, such as phylogenic tree creation for multiple disease states. As such, additional contributions are provided for multiple underlying states. Although these contributions are used for clustering immunosignaturing data with multiple underlying diseases, our approach of combining adaptive methodologies is novel and our proposed methods are flexible enough for adaptation to a variety of problems, even outside of the medical immunosignaturing area. The main contributions for this work are as follows:

- We propose a novel method for multi-state analysis and clustering that builds upon the corresponding single state methods, and we develop unique comparison criteria. The multi-state analysis is based upon the presumption that feature combinations are indicative of single disease states, but we also allow integration of the feature that combinations to indicate multiple states. Furthermore, the new combination of visible feature spaces and latent feature spaces allows us to capture cases of initial data with high variability and data with different distributions.

- We develop a flexible method based on the hierarchical Dirichlet process (HDP) that combines the visible features (from PCA or beta distribution fitting) and latent features (from BPFA) and then compares the overall features using a modified binary feature matrix comparison to enable adaptive clustering across multiple data groups. Direct application of the HDP to the multi-state features without the modified feature matrix does not yield good performance unless additional data processing steps are performed to extract the clustering results. This novel, enabling step also allows for the introduction of penalty or reward conditions for various comparisons that are application specific. This allows for flexibility across a variety of conditions where either strict user conditions create low tolerance for error, or where some variation is allowed for the sake of discovery. Additionally, this approach allows the novel determination of present states without the need for state significance thresholding specified by the user as well as reduced subjective user interaction.

- We develop the multi-state analysis for immunosignaturing data and key disease state data with flexible parameters to allow subsequent refinement and expansion of the algorithms linked to possible biological phenomena for better understanding. This intentional design allows the algorithms to be refined in a meaningful way as additional biological information is generated.

These main contributions are based on several proposed methods. The first method is an extension of the Z-PREDICT and ZB-PREDICT methods to allow for the comparison of $n > 1$ state possibilities. These methods are referred to as Z-PREDICTn (Z-matrix from Beta PDF REsolution and Discovery for Immunosignature Classification Testing up to n-states) and ZB-PREDICTn (Z-matrix from Beta PDF REolution and Discovery for Immunosignature Classification Testing up to n-states). While not adaptive, these methods allow for classification assuming that a number of underlying states is known [47]. In this method, the BPFA is provided with key state data and dataset information, and an average profile for each single state is constructed. The unknown state data is then compared to this average key data to achieve a ranking profile for each data entry and each state.

In order to eliminate the need to know underlying state information, we develop the H-PREDICT (HDP of PCA REsolution and Discovery for Immunosignature Classification Testing) and HB-PREDICT (HDP of Beta PDF REsolution and Discovery for Immunosignature Classification Testing) methods. In these cases, the state ranking profiles for each unknown data entry is classified using the HDP, allowing for common clustering criteria over each distinct ranking profile. It should be noted that without the ranking comparison described in Z-PREDICTn and ZB-PREDICTn, the direct application of the HDP or even the DP-GMM is not possible. This is because each immunosignature entry would still only be assigned to a single cluster.

1.3 Immunosignaturing Descriptions

A variety of datasets are used throughout this work to demonstrate the various algorithms. In order to show functionality for single state algorithms, a state is defined as a single disease and each data entry is the corresponding information, i.e. immunosignature, for each patient. In the case of multi-state algorithms, a state is still considered an underlying disease, but each patient (i.e. data entry) is assumed to have more than one underlying disease. Additionally, different methods require different definitions of classification success. Immunosignaturing data from the Arizona State University Biodesign Institute [14] was used to demonstrate the performance of each method. Median peptide intensity was used for all analysis. It should be noted that the disease selection is irrelevant for these data set constructions. Any combination of diseases could have been selected. As such, any deviations from these datasets is discussed in the corresponding results sections. Additionally, while there are roughly 10,000 unique sequences present in the array, they are often replicated at least once. As such, where applicable, datasets are reduced first by averaging together the median intensity values of repeated sequences.

1.3.1 Single Disease Dataset Descriptions

Microarray data sets for each patient derived from the CIM10K microarray template [9] were used. All methods described in this section were performed on several distinct datasets. Dataset 1 consists of 30 individuals with one of six disease states: *breast cancer, normal, glioma, cocci, sarcoma,* and *asthma post.* The patient order is according to the disease order just listed. No species or time point information is specified for these individuals. These are sometimes identified as *C1-C6* respectively on corresponding Dataset 1 tables. Dataset 2 consists of 25 individuals with one of five disease states: *Alzheimer's, asthma, influenza, Q-fever,* and *normal.* These are sometimes identified as *C1-C5* respectively on the corresponding Dataset 2 tables. Again, patients are placed in the dataset per the disease order previously indicated. No time point information is specified for these individuals. Individuals in the Alzheimer's, influenza and normal are from human samples, Q-fever individuals are mouse sample, and asthma samples are unspecified. No species information is used in the analysis and all immunosignatures are analyzed in the same manner. Median intensity values from the immunosignaturing microarrays are used, and all control sequences and machine mis-reads are removed from the datasets before analysis. Data types were chosen to represent a variety of disease states and even species where applicable, in order to show that no prior information is required other than the median intensity data to separate patient populations.

1.4 Report Organization

The dissertation is organized as follows. Chapter 2 provides background on the DP-GMM and the blocked Gibbs sampler, develops the PREDICT and B-PREDICT clustering algorithms and provides simulations of the algorithm performance. Chapter 3 presents theBPFA using the blocked Gibbs sampler and presents the Z-PREDICT and ZB-PREDICT algorithms with corresponding clustering results. The DP-GMM and BPFA are combined to form the BIO-PREDICT and BIOB-PREDICT clustering algorithms which are discussed in Chapter 4 with corresponding immunosignature clustering results. Chapter 5 proposes and demonstrates the performance of multistate clustering algorithms Z-PREDICTn and ZB-PREDICTn. The HDP is discussed in Chapter 6 and the H-PREDICT and HB-PREDICT clustering algorithms and their performances are presented. The performance of different proposed clustering algorithms is compared with the performance of a naive Bayes classifier in Chapter 7. Finally, in Chapter 8, conclusions and extensions to future work are discussed. A graphical depiction of the clustering algorithm progressions is given in Figure 1.2. The direction of the arrows demonstrates the flow of each algorithm, and the steps needed by the algorithm are given by the boxes the arrow line crosses. For example, BIO-PREDICT is obtained using PCA features, BPFA features, BPFA feature matrix analysis, and DP-GMM. The acronyms used in the dissertation are summarized in Table 1.1, and the acronyms we used for the proposed clustering algorithm names are defined in Table 1.2.

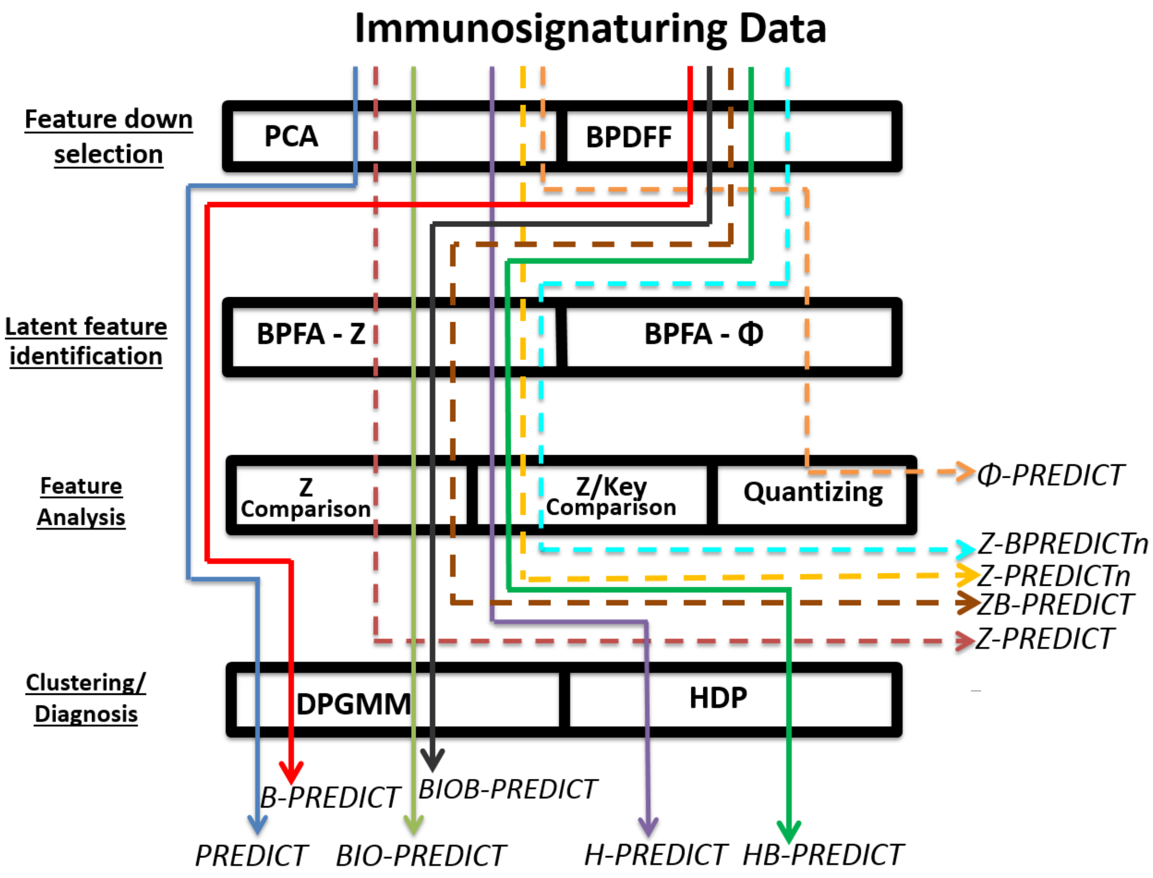


Figure 1.2: Summary depicting proposed clustering algorithms.

Table 1.1: Alphabetical list of acronyms used in this dissertation.

Acronym	Description
BP	Beta Process
BPF	Beta Probability Distribution Function Fit
BPFA	Beta Process Factor Analysis
CMA	Circular Moving Average
CRP	Chinese Restaurant Process
CRF	Chinese Restaurant Franchise
DP	Dirichlet Process
DP-GMM	Dirichlet Process Gaussian Mixture Model
GMM	Gaussian Mixture Model
HDP	Hierarchical Dirichlet Process
IBP	Indian Buffet Process
LOF	Left Ordered Form
LOOCV	Leave One Out Cross Validation
MCMC	Markov Chain Monte Carlo
MDD	Multi-Disease Dataset
MLE	Maximum Likelihood Estimator
PCA	Principle Component Analysis
PDF	Probability Distribution Function
RIC	Relaxed Immunosignaturing Clustering/Classification
SIC	Strict Immunosignaturing Clustering/Classification
SSLB	Single State Lower Bound
SSUB	Single State Upper Bound

Table 1.2: Alphabetical list of acronyms for proposed clustering algorithms.

Acronym	**Description**
BIO-PREDICT	**B**PFA **I**ncluding pr**O**cessing with **P**CA **RE**solution and **D**P-GMM for **I**mmunosignature **C**lassification **T**esting
BIOB-PREDICT	**B**PFA **I**ncluding pr**O**cessing with **B**eta**P**DF **RE**solution with **D**P-GMM for **I**mmunosignature **C**lassification **T**esting
B-PREDICT	**B**eta**P**DF **RE**solution with **D**P-GMM for **I**mmunosignature **C**lassification **T**esting
HB-PREDICT	**H**DP of **B**eta **P**DF **RE**solution and **D**iscovery for **I**mmunosignature **C**lassification **T**esting
H-PREDICT	**H**DP of **P**CA **RE**solution and **D**iscovery for **I**mmunosignature **C**lassification **T**esting
PREDICT	**P**CA **RE**solution with **D**P-GMM for **I**mmunosignature **C**lassification **T**esting
ZB-PREDICT	**Z**-matrix from **B**eta **P**DF **RE**solution and **D**iscovery for **I**mmunosignature **C**lassification **T**esting
ZB-PREDICTn	**Z**-matrix from **B**eta **P**DF **RE**solution and **D**iscovery for **I**mmunosignature **C**lassification **T**esting up to **n**-states
Z-PREDICT	**Z**-matrix from **P**CA **RE**solution and **D**iscovery for **I**mmunosignature **C**lassification **T**esting
Z-PREDICTn	**Z**-matrix from **P**CA **RE**solution and **D**iscovery for **I**mmunosignature **C**lassification **T**esting up to **n**-states
Φ-PREDICT	**Φ** **P**CA **RE**solution and **D**iscovery for **I**mmunosignature **C**lassification **T**esting

Chapter 2

FEATURE SELECTION AND ADAPTIVE CLUSTERING USING DIRICHLET PROCESS GAUSSIAN MIXTURE MODELING

Processing immunosignature data can be computationally intensive due to the large number of spots on a single array and the fact that there is at least one microarray sample for each patient of interest. Immunosignature microarrays are designed to have a large number of random peptide sequences; the CIM10K array has 10,000 sequences [9] and the most current immunosignature microarray technology has 330,000 sequences [14]. As a result, we want to reduce processing complexity without unnecessarily losing the relationship between patients and disease state. It is also important to develop a processing method that does not require significant patient, disease state, immunosignaturing, or microarray knowledge for accurate performance. We consider two feature models for data reduction and discernible feature space mapping: principal component analysis (PCA) and beta probability density function fitting (BPF). Once features that depend on disease states are extracted, clustering can be performed to group patients according to disease state. We specifically consider the Dirichlet process Gaussian mixture model approach to design an adaptive, unsupervised clustering method without requiring *prior* training data.

2.1 Principal Component Analysis

The PCA method was previously used for immunosignature feature extraction to reduce the overall data dimensionality of the median peptide intensities [9, 10]. PCA is a general technique that seeks to approximate a signal by removing redundancy, retaining only essential signal properties [49, 50]. The immunosignaturing data has high dimensionality ($\sim 10^4$) and the number of patient samples ($\sim$ 5 to 100) is much smaller than the dimensionality. Due to the large difference between data dimensionality and samples, a modified covariance estimate is required to achieve a robust PCA representation [51]. Without this approximation, PCA would result in high condition numbers and large estimation errors.

Given a vector $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_N]$ of N data points the PCA representation is given by [52, 53]:

$$\mathbf{x} = \boldsymbol{\mu}_{\mathbf{x}} + \boldsymbol{\Lambda}\boldsymbol{\Gamma} + \boldsymbol{\epsilon} \tag{2.1}$$

where $\boldsymbol{\Gamma}$ is a matrix of factor parameters, $\boldsymbol{\Lambda}$ is a vector of the eigenvalues $\lambda_1, \lambda_2, \dots$, $\boldsymbol{\mu}_{\mathbf{x}}$ is a vector of data means, and $\boldsymbol{\epsilon}$ is the error vector. PCA generally requires that the mean vector is removed before the decomposition. The model assumes that [52]:

$$E[\mathbf{x}] = E[\boldsymbol{\epsilon}] = E[\boldsymbol{\Gamma}] = 0. \tag{2.2}$$

$E[\cdot]$ denotes the expectation operator. The covariance matrix $\mathbf{C}_{\mathbf{x}}$ of the zero-mean data is given by:

$$\mathbf{C}_{\mathbf{x}} = \boldsymbol{\Lambda}\boldsymbol{\Lambda}^T + \mathbf{C}_{\boldsymbol{\Delta}} \tag{2.3}$$

where $\mathbf{C}_{\boldsymbol{\Delta}} = E[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T]$ is the error covariance matrix. For very large N, a large covariance matrix needs to be reliably calculated. The sampled version does not provide a good estimate of the covariance matrix of $\mathbf{x}$ due to the large data dimensionality. Instead, we use an improved covariance estimate using sample shrinkage. The estimate of the i, jth element of the covariance matrix, $i, j = 1 \dots N$ is given by [51]:

$$\hat{C}_{i,j} = \begin{cases} \sigma_{i,i} & i = j \\ \hat{r}_{i,j}\sqrt{\sigma_{i,i}\sigma_{i,j}} & i \neq j \end{cases} \tag{2.4}$$

where $\hat{r}_{i,j}$ is the correlation estimate and ρ is the estimated shrinkage intensity [51]:

$$\hat{r}_{i,j} = \begin{cases} 1 & i = j \\ r_{i,j} \min(1, \max(0, 1 - \rho)) & i \neq j \end{cases} \tag{2.5a}$$

$$\rho = \frac{\sum_{i \neq j} \sigma^2_{r_{i,j}}}{\sum_{i \neq j} r^2_{i,j}} \tag{2.5b}$$

$\sigma_{r_{i,j}}$ is the estimated variance of sample correlation $r_{i,j}$, $s_{i,j}$ is the sample variance:

$$s_{i,j} = \frac{1}{\kappa - 1} \sum_{k=1}^{\kappa} (x_{k,i}) - \bar{x}_j \tag{2.6a}$$

$$\bar{x}_i = \frac{1}{\kappa} \sum_{k=1}^{\kappa} x_{k,i} \tag{2.6b}$$

where $\bar{x}_i$ is the sample mean and $x_{k,i}$ is the kth observation of x_i, $k = 1, \ldots, \kappa$. If we normalize x_i to have zero mean and unit variance, we can obtain the estimate variance of the correlation in equation (2.5b) as [54]:

$$\sigma^2_{r_{i,j}} = \frac{\kappa}{(\kappa - 1)^3} \sum_{k=1}^{\kappa} (w^*_{k,j,i} - w^*_{i,j})^2 \tag{2.7}$$

where:

$$r_{i,j} = \frac{\kappa}{\kappa - 1} w^*_{i,j}, \tag{2.8a}$$

$$w^*_{i,j} = \frac{1}{\kappa} \sum_{k=1}^{n} w^*_{k,i,j}, \tag{2.8b}$$

$$w^*_{k,i,j} = (x^*_{k,i} - x^*_i)(x^*_{k,j} - x^*_j) \tag{2.8c}$$

$$x^*_i = \frac{1}{\kappa} \sum_{k=1}^{n} x^*_{k,i} \tag{2.8d}$$

When applying this modified covariance estimate, it is possible to produce a more robust PCA representation that contains more accurate eigenvalues [51]. Once the data dimensionality is reduced, the log of the PCA components is taken in order to address the scale discrepancy between dimensions. The reduced data dimensionality is found by performing eigenvalue analysis and keeping only components with sufficiently large eigenvalues. It is up to the researcher to determine what percentage of the original data is kept. This percentage can be calculated by dividing the sum of the kept eigenvalues over the total of all eigenvalues. For the immunosignature data, the number of unique peptide microarray spots per patient and the reduced dimensionality is the median intensity of the microarray spots mapped to the log PCA domain (i.e. log of the linear combination of the median intensity of the microarray spots). Note that the log was used to help with feature space separation.

2.2 Beta Probability Density Function Fitting

The second method for feature extraction is based on fitting microarray binding intensity data to the parameters of beta probability density functions (PDFs) [57–61]. The beta PDF fitting (BPF) approach uses maximum likelihood to estimate the beta PDF parameters that best fit a microarray data sample [55, 56]. These beta distribution parameters are used to define the dimensionality and encode the immunosignaturing behavior in a limited number of parameters. This is also a representation of the visible process features (median intensity distributions) that the user can discern by data examination. The two beta distribution parameters α and β create the two-dimensional space, effectively reducing the dimensionality from ∼10,000 to 2. While this is a dramatic dimension reduction, it provides a good first estimate of the intensity behaviors between the different populations without running individual statistics between all of the groups and without individually investigating every pep-

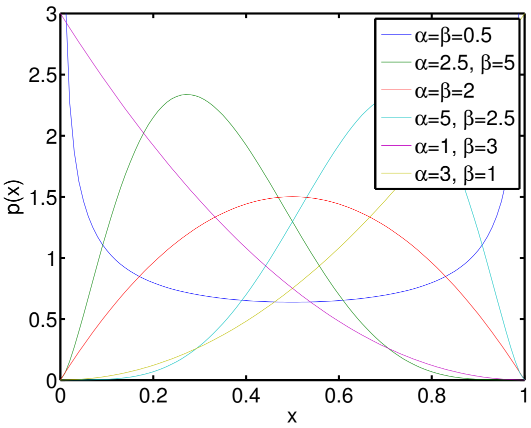


Figure 2.1: Example of beta probability density functions.

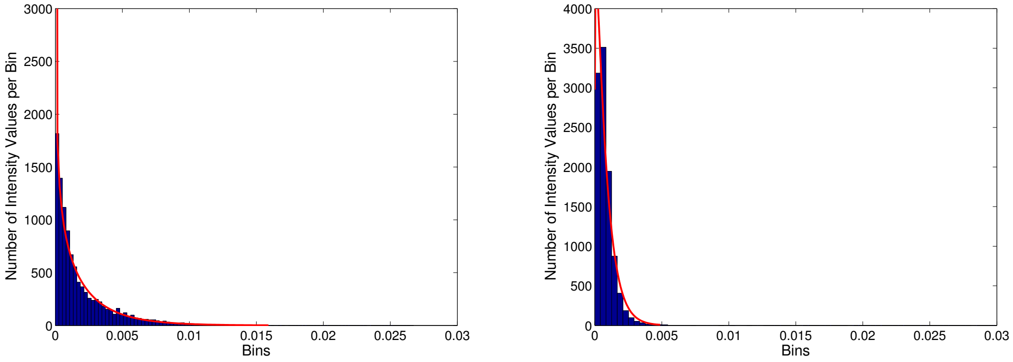
tide comparison between groups. Two example histograms are given in Figure 2.2, one for a patient with breast cancer and one with normal pathology, with beta PDFs fitted for each patient using un-normalized median intensity immunosignature data. Note that in the algorithm the data is normalized for patients so that the intensity values are between 0 and 1. This is required in order to use the beta PDF, which is only defined between 0 and 1. The beta PDF is selected because of the diverse PDF shapes that may be described by its α and β parameters, as seen in Figure 2.1. The beta PDF for the nth patient, $n = 1, \ldots, N$ is given by [62]:

$$\beta(x_{l,n}; \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{(\Gamma(\alpha)\Gamma(\beta))} \, {x_{l,n}}^{\alpha-1}(1 - x_{l,n})^{\beta-1} \tag{2.9}$$

where $x_{l,n}$, $l = 1, \ldots, D$ is the lth microarray data value for the the nth patient.

The beta PDF parameters that best fit the data of the n patient are found using maximum likelihood estimation (MLE) as:

$$\{\hat{a}_n, \hat{b}_n\} = \arg\max_{\alpha,\beta} \prod_{l=1}^{D} \beta(x_{l,n}; \alpha, \beta) \,, \tag{2.10}$$



(a) Histogram of the intensity values for a patient with breast cancer and the beta PDF that best fits the data.

(b) Histogram of the intensity values for a normal patient with and the beta PDF that best fits the data.

Figure 2.2: Beta PDFs fit to histograms of patient immunosignaturing data.

where α and β are the PDF parameters. This can be extended to the N patients within the data set such that there are now vectors of parameters that fully describe the set of immunosignaturing datasets for various patients, where each dataset has a single $\hat{\alpha}_n$ and $\hat{\alpha}_n$ value found using MLE. Modeling the distribution of D-dimensional normalized binding intensities $\mathbf{x}_n = [x_{1,n}, x_{2,n}, \ldots, x_{D,n}]$ for the nth patient, $n = 1, \ldots, N$, we can obtain the PDF parameter vectors:

$$\boldsymbol{\alpha} = [\hat{\alpha}_1, \hat{\alpha}_2 \ldots, \hat{\alpha}_N], \tag{2.11a}$$

$$\boldsymbol{\beta} = [\hat{\beta}_1, \hat{\beta}_2 \ldots, \hat{\beta}_N] \tag{2.11b}$$

The resulting output is that the nth patient is characterized by the two beta PDF parameters $\hat{\alpha_n}$ and $\hat{\beta_n}$ describing the beta PDF created by its median intensities.

2.3 DP-GMM Clustering of Immunosignatures

In order to facilitate adaptive clustering of immunosignature data, we model data features using Dirichlet process Gaussian mixture models. Modeling features reduces the large dimensionality of the immunosignatures and thus the computational cost of the clustering algorithm; it also provides insight into the relationships between patients with similar diseases.

2.3.1 Conjugate Priors

Before discussing the DP-GMM clustering algorithm, we first provide some background on conjugate priors from a Bayesian setting, on Dirichlet processes and on Gaussian mixture models. If the posterior distributions $p(\theta|x)$ are in the same family as the prior probability distribution p(theta), the prior and posterior are then called conjugate distributions, and the prior is called a conjugate prior for the likelihood function. Markov chain Monte Carlo MCMC methods and blocked Gibbs sampling algorithms are built upon the premise of conjugate priors. This views parameter estimation as a Bayesian inference problem where the posterior PDFs are estimated based on data used for estimation [63]. This exploits a conjugate prior relationship that explicitly describes these parameter relationships in a Bayesian sense. Considering a random data vector $\mathbf{x}$ and a random parameter vector $\boldsymbol{\psi}$, and assuming similar statistical characteristics for the posterior PDF $p(\boldsymbol{\psi}|\mathbf{x})$ and the prior PDF $p(\boldsymbol{\psi})$, then the posterior and prior PDFs are considered conjugate distributions. The prior PDF is then the conjugate prior for the likelihood function $p(\mathbf{x}|\boldsymbol{\psi})$ [63–65]. The relationship between these PDFs is given by Bayes theorem as:

$$p(\boldsymbol{\psi}|\mathbf{x}) = \frac{p(\mathbf{x}|\boldsymbol{\psi})p(\boldsymbol{\psi})}{p(\mathbf{x})} \tag{2.12}$$

This relationship is often represented as:

$$p(\boldsymbol{\psi}|\mathbf{x}) \propto p(\mathbf{x}|\boldsymbol{\psi})p(\boldsymbol{\psi}) \tag{2.13}$$

The term *hyperparameter* is also used in the context of conjugate priors. Hyperparameters are PDF parameters that have their own prior distributions and can be estimated using MCMC methods [3]. Essentially, it is a distribution over the parameters of a particular distribution. Initial distributions with conjugate priors are chosen for their known relationships and ease of implementation to create more efficient algorithms.

2.3.2 Dirichlet Process and Gaussian Mixture Modeling

Using conjugate priors, we discuss next Gaussian mixture modeling (GMM) and the Dirichlet process (DP). The DP-GMM is used in a variety of applications [66–69] to model data distributions using an unlimited number of mixture components [70]. Given a data or feature vector $\mathbf{x}$, a mixture model is described by the PDF

$$p\left(\mathbf{x}|\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_M\right)=\sum_{m=1}^{M} w_m\, p\left(\mathbf{x};\boldsymbol{\theta}_m\right), \tag{2.14}$$

where $\{w_1,\ldots,w_M\}$ are the individual mixture component weights, $\boldsymbol{\theta}_m$ is the parameter space representing the PDF $p(\mathbf{x}|\boldsymbol{\theta_m})$, and M is the number of mixture components [71]. The goal of the mixture model is to estimate M, w_m, and $\boldsymbol{\theta}_m$, $m=1,\ldots,M$ that best fit this data. The DP-GMM provides an adaptive approach to determine cluster model parameters [72] where an infinite number of mixture components and weighting factors are theoretically possible [73]. In this case, $p(\mathbf{x};\boldsymbol{\theta}_m)\sim\mathcal{N}(\mathbf{x};\mu_m,\Sigma_m)$ is a Gaussian PDF with a parameter vector consisting of the mean and covariance of the PDF, $\boldsymbol{\theta}_m=\{\boldsymbol{\mu}_m,\boldsymbol{\Sigma}_m\}$. Therefore, the mixture model can be specified with Gaussian distributions such that a complete representation of clustering in the space is:

$$p\left(\mathbf{x}|\mathbf{w},\boldsymbol{\mu},\boldsymbol{\Sigma}\right)=\sum_{m=1}^{M} w_m\,\mathcal{N}\left(\mathbf{x};\boldsymbol{\mu}_m,\boldsymbol{\Sigma}_m\right). \tag{2.15}$$

In a Bayesian sense, a prior PDF must be selected in order to determine the nonparametric statistics related to a dataset. The DP is a prior in which a base distribution, G_0, and an innovation parameter α, fully describe the DP [71]. The innovation parameter effectively describes how likely a new data point is to be placed within a cluster of previous data or in a newly formed cluster. In the *stick-breaking* algorithm of the DP [74], this corresponds to how fine the breaks or cluster divisions are made. Consider the distribution [75]:

$$\begin{aligned} G &\sim \mathrm{DP}(\alpha, G_0), & \text{(2.16a)} \\ \boldsymbol{\theta}_n | G &\sim G, \;\; n = 1, \ldots, N. & \text{(2.16b)} \end{aligned}$$

that is drawn from a DP with innovation parameter α and base distribution G_0 with $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N\}$. Then, G_0 is the expected value of G and α determines how close G is to G_0. In particular, G is discrete and has the following stick-breaking representation [74]:

$$\begin{aligned} &\boldsymbol{\theta}_m \sim G_0, \quad m = 1, \ldots, \infty \\ &v_i \sim \mathrm{Beta}(1, \alpha), \quad i = 1, \ldots, \infty \\ &w_m = v_m \prod_{i=1}^{m-1} (1 - v_i), \quad m = 1, \ldots, \infty \\ &G(\boldsymbol{\theta}) = \sum_{m=1}^{\infty} w_m \; \delta(\theta - \theta_m) \, . \end{aligned} \tag{2.17}$$

where where $\delta(\cdot)$ is the Kronecker delta function. This designation comes from the idea that a unit length stick may be broken such that the size of each successive break is representative of $w_m = \Pr(\boldsymbol{\theta} = \boldsymbol{\theta}_m)$. To understand the assignment to a particular cluster $\boldsymbol{\theta}_n$, it is possible to integrate out G, thus describing the Pólya urn relation [70, 76–79] given by:

$$p(\boldsymbol{\theta}_n|\boldsymbol{\Theta}^{(-n)}, \alpha, G_0) = \frac{1}{\alpha + N - 1} \sum_{\substack{n'=1 \\ n' \neq n}}^{N} \delta(\boldsymbol{\theta}_n - \boldsymbol{\theta}_m) + \frac{\alpha}{\alpha + N - 1} G_0(\boldsymbol{\theta}_n), \tag{2.18}$$

where $\boldsymbol{\Theta}^{(-n)}$ are the parameters except for $\boldsymbol{\theta}_n$, and $N_m^{(-n)}$ is the number of variables in $\boldsymbol{\Theta}^{(-n)}$ that are equal to $\boldsymbol{\theta}_m$. This representation may be rewritten as:

$$p(\boldsymbol{\theta}_n|\boldsymbol{\Theta}^{(-n)}, \alpha, G_0) = \frac{1}{\alpha + N - 1} \sum_{m=1}^{M} N_m^{(-n)} \, \delta(\boldsymbol{\theta}_n - \boldsymbol{\theta}_m) + \frac{\alpha}{\alpha + N - 1} G_0(\boldsymbol{\theta}_n) \tag{2.19}$$

This further helps to illustrate the functionality of the innovation parameter, α. In particular, the probability of choosing an existing cluster value is given as $\Pr(\boldsymbol{\theta}_m = \boldsymbol{\theta}_n) = N_m^{(-n)}/(\alpha + N - 1)$, and the probability of selecting a new cluster value is given by $\Pr(\boldsymbol{\theta}_m \neq \boldsymbol{\theta}_n) = \alpha/(\alpha + N - 1)$. This then leads to the DP-GMM representation as described in Equation (2.15) whose corresponding stick-breaking process representation is given as:

$$\begin{aligned}
\boldsymbol{\theta}_m &\sim G_0, \;\; m = 1, \ldots, \infty, && \text{(2.20a)} \\
v_k &\sim \text{Beta}(1, \alpha), \;\; k = 1, \ldots, \infty, && \text{(2.20b)} \\
w_m &= v_m \prod_{k=1}^{m-1} (1 - v_k), \;\; m = 1, \ldots, \infty, && \text{(2.20c)} \\
c_n|\mathbf{w} &\sim \text{Categorical}(\mathbf{w}), \;\; n = 1, \ldots, N, && \text{(2.20d)} \\
\mathbf{x}_n|c_n &\sim p(\mathbf{x}_n|\boldsymbol{\theta}_{c_n}), \;\; n = 1, \ldots, N. && \text{(2.20e)}
\end{aligned}$$

The variable c_n indicates which of the M possible clusters includes $\mathbf{x}_n$ and categorical refers to the assignment to one of the M possible clusters. While in theory the DPGMM is infinite, a practical truncation limit may be selected such that the truncated representation is a close approximation of the infinite case, and it can also help to simplify the numerical calculations for approximating the DP [80]. This limit M, can be set by the user or it can be calculated directly. The truncation error is related to M and is given by [81]:

$$4N \exp(-(M-1)/\alpha). \tag{2.21}$$

In actuality, the value of M may be larger than the true number of latent clusters for a particular data set. Additionally, if M is calculated, then it can be adjusted by the innovation parameter α, as in Equation (2.21).

2.3.3 The Dirichlet Process and Blocked Gibbs Sampler

This conjugate prior relationship is used extensively to simplify calculations for posterior distributions estimated using the blocked Gibbs sampler algorithm. Using an MCMC technique such as the blocked Gibbs sampler, it is possible to iteratively estimate posterior distribution parameters [82]. Considering the mixture model given in Equation (2.14) and using the notation of Equation (2.20), the blocked Gibbs sampler, at the ith iteration in the Markov chain estimates [80, 82]:

$$\boldsymbol{\theta}_m^{(i)} \sim p\,(\boldsymbol{\theta}_m|\mathbf{c}^{(i-1)}, \mathbf{x}_n),\; m = 1, \ldots, M, \tag{2.22a}$$

$$c_n^{(i)} \sim p\,(c_n|\boldsymbol{\Theta}^{(i)}, \mathbf{w}^{(i-1)}, \mathbf{x}_n),\; n = 1, \ldots, N, \tag{2.22b}$$

$$w_m^{(i)} \sim p\,(w_m|\mathbf{c}^{(i)}),\; m = 1, \ldots, M. \tag{2.22c}$$

These can be expressed in terms of conjugate prior relationships [80]:

$$
\begin{aligned}
p\left(\boldsymbol{\theta}_m|\mathbf{c},\mathbf{x}_n\right) &\propto G_0(\boldsymbol{\theta}_m)\prod_{n:c_n=m} p\left(\mathbf{x}_n|\boldsymbol{\theta}\right),\, m=1,\ldots,M, && (2.23)\\
p\left(c_n|\boldsymbol{\Theta},\mathbf{w},\mathbf{x}_n\right) &= \sum_{m=1}^{M}\Big(w_m\, p\left(\mathbf{x}_n|\boldsymbol{\theta}_m\right)\Big)\,\delta(c_n-m),\, n=1,\ldots,N, && (2.24)\\
p\left(w_m|\mathbf{c}\right) &= v_m\prod_{j=1}^{m-1}(1-v_j),\, m=1,\ldots,M && (2.25)
\end{aligned}
$$

where v_m is also defined as:

$$v_m \sim \text{Beta}\Big(1+N_m^*, \alpha+\sum_{m'=m+1}^{M} N_{m'}^*\Big)\,, \tag{2.26}$$

and $n : c_n = m$ denotes the indices in $\mathbf{c}$ such that $c_n = m$, and N_m^* is the number of elements in $\mathbf{c}$ that are equal to m.

While this describes the conjugate prior relationship, the specific mathematical equations for the BGS execution require the selection of the prior and likelihood distributions. In the case of the DP-GMM, the likelihood distribution is Gaussian where as the prior distribution, G_0 is Normal-Wishart. The Normal-Wishart PDF is used because this is the multidimensional generalization for the DPGMM. Thus, the relationship between the prior and posterior distributions may be described by [63]:

$$G_0(\boldsymbol{\theta}) \triangleq \mathcal{NW}\left(\boldsymbol{\mu},\boldsymbol{\Sigma}^{-1}\,;\,\boldsymbol{\mu}_{\mathcal{N}},\tau_{\mathcal{N}},\xi_{\mathcal{W}},\iota_{\mathcal{W}}\right), \tag{2.27a}$$

$$p\left(\boldsymbol{\theta}|\mathbf{c},\mathbf{X}\right) \triangleq \mathcal{NW}\left(\boldsymbol{\mu},\boldsymbol{\Sigma}^{-1}\,;\,\tilde{\boldsymbol{\mu}}_{\mathcal{N}},\tilde{\tau}_{\mathcal{N}},\tilde{\xi}_{\mathcal{W}},\tilde{\iota}_{\mathcal{W}}\right) \tag{2.27b}$$

The hyperparameters $\boldsymbol{\mu}_{\mathcal{N}}$, $\tau_{\mathcal{N}}$, $\xi_{\mathcal{W}}$, $\iota_{\mathcal{W}}$, $\tilde{\boldsymbol{\mu}}_{\mathcal{N}}$, $\tilde{\tau}_{\mathcal{N}}$, $\tilde{\xi}_{\mathcal{W}}$, $\tilde{\iota}_{\mathcal{W}}$ are described by [63]:

$$\tilde{\boldsymbol{\mu}}_{\mathcal{N}} = \frac{\tau_{\mathcal{N}}\,\boldsymbol{\mu}_{\mathcal{N}} + N\boldsymbol{\mu}_{\mathbf{x}}}{\tau_{\mathcal{N}} + N}\,, \tag{2.28}$$

$$\tilde{\tau}_{\mathcal{N}} = \tau_{\mathcal{N}} + N\,, \tag{2.29}$$

$$\tilde{\iota}_{\mathcal{W}} = \iota\mathcal{W} + \boldsymbol{\Sigma}_{\mathbf{x}} + \frac{\tau_{\mathcal{N}}\,N}{\tau_{\mathcal{N}} + N}\left(\boldsymbol{\mu}_{\mathcal{N}} - \boldsymbol{\mu}_{\mathbf{x}}\right)\left(\boldsymbol{\mu}_{\mathcal{N}} - \boldsymbol{\mu}_{\mathbf{x}}\right)^{T}, \tag{2.30}$$

$$\tilde{\xi}_{\mathcal{W}} = \xi_{\mathcal{W}} + N\,, \tag{2.31}$$

and $\boldsymbol{\mu}_{\mathbf{x}}$ and $\boldsymbol{\Sigma}_{\mathbf{x}}$ are the mean and covariance of $\mathbf{X}$, and $\boldsymbol{\mu}_{\mathcal{N}}$, $\tau_{\mathcal{N}}$, $\xi_{\mathcal{W}}$, and $\iota_{\mathcal{W}}$ are user specified with the restrictions:

$$\tau_{\mathcal{N}} > 0, \tag{2.32a}$$

$$\xi_{\mathcal{W}} > D - 1 \tag{2.32b}$$

where D is the number of dimensions in $\mathbf{x}$. The algorithm and further description of the update equations for the implementation of the blocked Gibbs sampler are provided in Algorithm 1.

Gaussian mixture modeling in the Bayesian sense is described as using a Gaussian prior to probabilistically describe data subgroup behavior within an overall data population. The mixture of subgroup distributions defines group membership without providing group identification. The DP is one such method of yielding subgroup clusters, and it may rely on various distributions to describe the data behavior by way of features [73]. In the immunosignaturing case, the distributions correspond to the normalized binding intensities, which may have a variety of distribution shapes. The learning of the associated distribution parameters can be done via recursive estimation through construction of a Markov chain.

2.4 DP-GMM Immunosignature Clustering Algorithms

2.4.1 DP-GMM Clustering with PCA Features

While the DPGMM is an effective way to adaptively cluster various data sets, it can be computationally intensive. Thus, the PREDICT, or **P**CA **RE**solution with **D**PGMM for **I**mmunosignature **C**lassification **T**esting is presented to help analyze immunosignaturing data. PREDICT uses the log results of principal component analysis (PCA) for dimensionality reduction followed by the DPGMM for classification. In this way, patients are clustered with other patients sharing similar feature characteristics, in this case PCA features. One underlying assumption for this method is that the patient will have only a single disease pathology. This is due to the fact that the DPGMM will assign each patient to a single cluster during each iteration of the blocked Gibbs sampler. While average values are computed based on a user specified number of sample iterations, the end result is still a single cluster assignment.

2.4.2 DP-GMM Clustering with Beta PDF Fitting Features

There are alternative feature reduction schemes to PCA, including beta PDF fitting, as discussed here. The B-PREDICT, or **B**eta **P**DF **RE**solution with **D**PGMM for **I**mmunosignature **C**lassification **T**esting, algorithm uses Beta PDF fitting to reduce the immunosignaturing feature space, and then feeds a scaled result of that into the DPGMM. The scaling is left to the user, and is simply available to avoid a small numbers problem when clustering. In this manner, B-PREDICT is identical to PREDICT, except that it uses Beta PDF fitting to reduce the feature space, and as such requires less computations than the PCA based method (e.g. for a modified covariance matrix), and the feature space will always be two dimensions.

2.5 DP-GMM Clustering Results

2.5.1 Strict and Relaxed Immunosignaturing Clustering Criteria

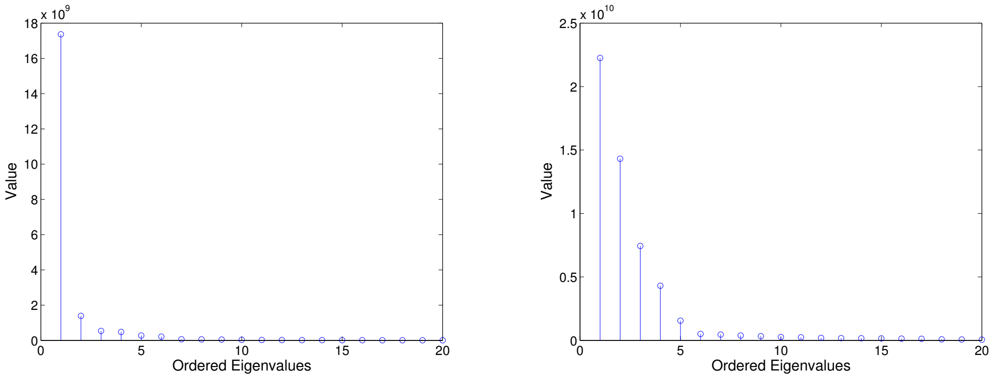
Before discussing the results for each of the methods, it is helpful to define what is considered correct or successful classification. When dealing with biological disease data amongst patients, it is possible that different patients, although having the same disease, will have different responses to diseases. Physically this may correspond to a range of symptoms, their severity, their progression over time, etc. For example those infected with influenza may experience a combination of a variety of symptoms [83]. While each patient may have a slightly different disease expression for a particular disease state, in general it is possible to group individuals by these expressions. However, this may result in multiple groupings that describe the same disease state. As such, we discuss a strict and a relaxed definition of immunosignaturing clustering success.

In the strict immunosignaturing clustering/classification definition (SIC), only one group may exist per disease state and any individual falling outside the group whose ground truth would indicate the same disease will be seen as a misclassification. In the relaxed immunosignaturing clustering/classification (RIC) definition, multiple groups may be used to represent a single disease as long as two or more individuals are present in this additional group. However, it will be considered a misclassification if individuals from another disease state are incorrectly placed in this group. It is believed that these two definitions will offer some flexibility to compare the results from the various methods, but without allowing various algorithm parameters to be set such that each patient is considered a new group. In some cases it may be more appropriate to use the SIC definition of classification success if no underlying relationship indicating multiple groupings is available. For example, a training set of data may be used to help determine if multiple groups are present for a particular disease state. In cases where an underlying relationship is present, it may be more appropriate to use the RIC definition of classification success.

2.5.2 Results Using the PREDICT Clustering Algorithm

In order to demonstrate the clustering performance of the DP-GMM based algorithms, we use the datasets described in Chapter 1. As previously stated, the dimensionality in this case is defined by the PCA of the microarray median intensity data assuming that the dimensionality is defined by the number of peptide sequences. After PCA, dimensionality is defined as linear combinations of peptide intensity results determined by the significant eigenvalues from the PCA. The PCA algorithm was applied as in equations (2.1) and (2.4) and the log was taken as the input to the DPGMM. At minimum, the first three principle components were selected for Dataset 1, which represents 93.5% of the data, and the first five principal components were used from Dataset 2 which represents 93.5% of the data. Further dimensions may be added if desired from the PCA results, but the number of components was selected based on the plots of the eigenvalues as shown in Figure 2.3.

The confusion matrices for Dataset 1 with diseases identified as C1-C6 may be found in Table 2.1 and 2.2. The innovation parameter α was set to 15, the error was set to 5x10^{-1}, which produced a truncation of $M = 83$. For the blocked Gibbs sampler, 2000 iterations were done for burn-in and 2000 sample iterations were then performed. The dataset 1 PREDICT results of the clustering can be seen in Figure 2.4. The true disease states are color coded for viewing ease, and the bar heights indicate class assignments by the algorithm. Another representation is given in the same figure to help show how bar heigh relates to the clustering results where each face represents a patient, and the color is indicative of the ground truth disease

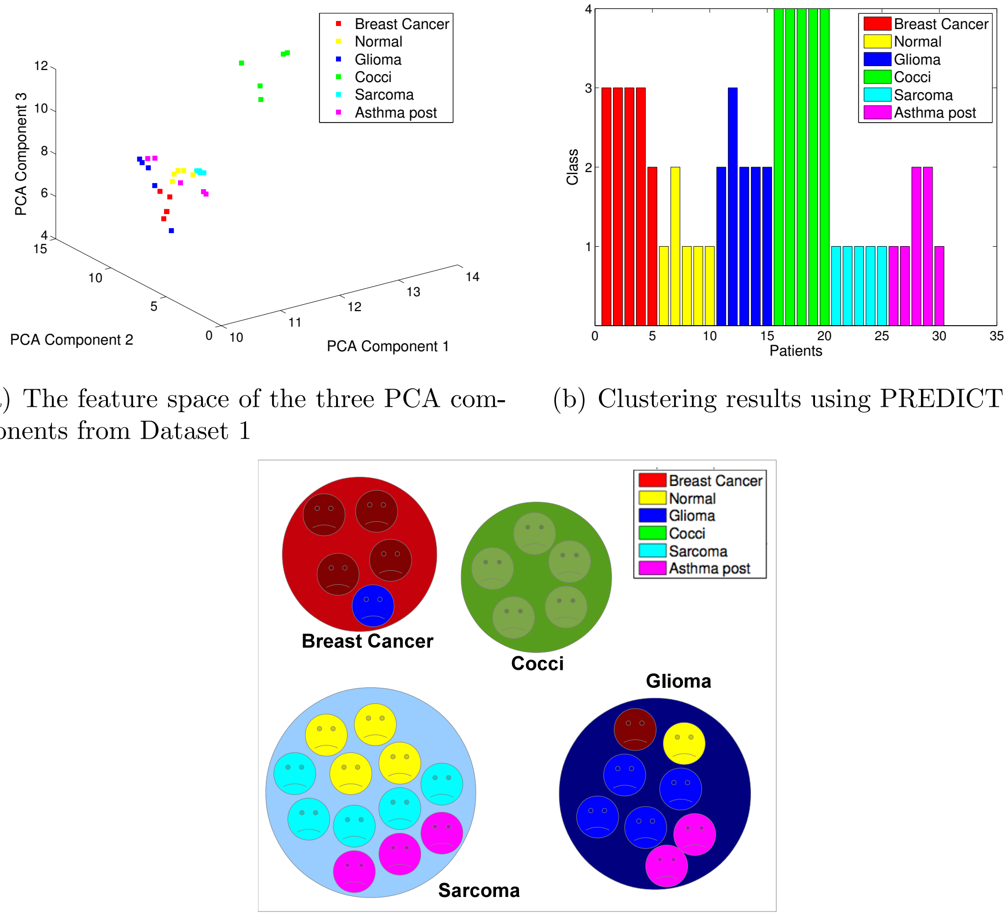


(a) Eigenvalue plot from Dataset 1.

(b) Eigenvalue plot from Dataset 2.

Figure 2.3: The first 20 significant eigenvalues from PCA analysis of each dataset.

information. In this depiction, patient color represents the underlying disease and cluster color represents the corresponding cluster for each disease. Instances with matching patient and cluster colors are correct clustering and instances with different colors are incorrect clustering. The correct classification for both SIC and RIC was 60%. The confusion matrices for Dataset 2 with diseases identified as C1-C5 may be found in Table 4.3 and 4.4. The innovation parameter α was set to 35, the error was to 1x10^{-4}, which results in $M = 485$. For the BGS, the number of burn-in iterations was set to 2000 and the number of sample iterations was set to 3000. The dataset 2 PREDICT results can be seen in Figure 2.5; the SIC and RIC results were both 64%.



(a) The feature space of the three PCA components from Dataset 1

(b) Clustering results using PREDICT.

(c) Clustering results using PREDICT.

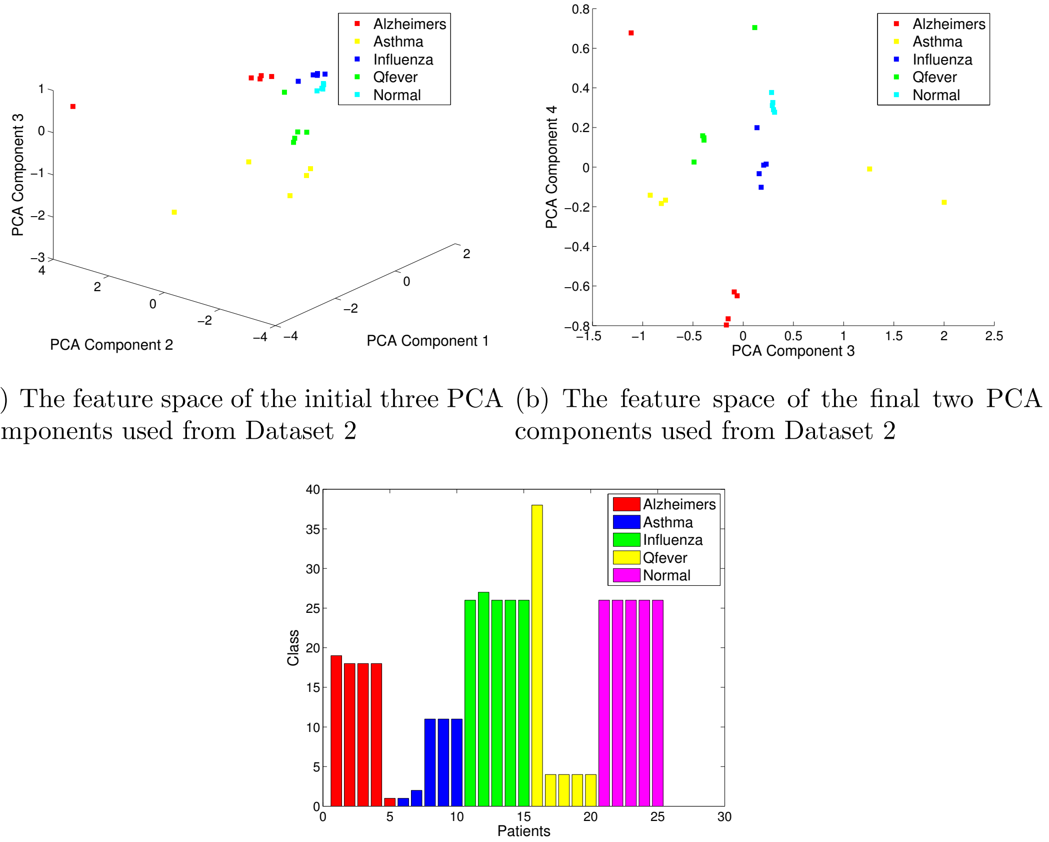
Figure 2.4: PREDICT results for Dataset 1

Class	C_1	C_2	C_3	C_4	C_5	C_6
C_1	**4/5**	0/5	1/5	0/5	0/5	0/5
C_2	0/5	**0/5**	1/5	0/5	4/5	0/5
C_3	1/5	0/5	**4/5**	0/5	0/5	0/5
C_4	0/5	0/5	0/5	**5/5**	0/5	0/5
C_5	0/5	0/5	0/5	0/5	**5/5**	0/5
C_6	0/5	0/5	2/5	0/5	3/5	**0/5**

Table 2.1: Dataset 1 PREDICT SIC confusion matrix.

Class	C_1	C_2	C_3	C_4	C_5	C_6
C_1	**4/5**	0/5	1/5	0/5	0/5	0/5
C_2	0/5	**0/5**	1/5	0/5	4/5	0/5
C_3	1/5	0/5	**4/5**	0/5	0/5	0/5
C_4	0/5	0/5	0/5	**5/5**	0/5	0/5
C_5	0/5	0/5	0/5	0/5	**5/5**	0/5
C_6	0/5	0/5	2/5	0/5	3/5	**0/5**

Table 2.2: Dataset 1 PREDICT RIC confusion matrix.



(a) The feature space of the initial three PCA components used from Dataset 2

(b) The feature space of the final two PCA components used from Dataset 2

(c) Classification results using PREDICT

Figure 2.5: PREDICT results for Dataset 2

Class	C_1	C_2	C_3	C_4	C_5
C_1	**3/5**	0/5	0/5	0/5	0/5
C_2	0/5	**3/5**	0/5	0/5	0/5
C_3	0/5	0/5	**1/5**	0/5	4/5
C_4	0/5	0/5	0/5	**4/5**	0/5
C_5	0/5	0/5	0/5	0/5	**5/5**

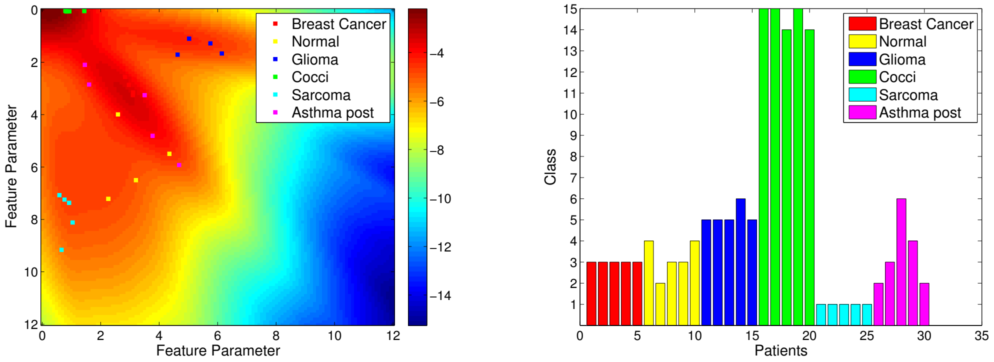
Table 2.3: Dataset 2 PREDICT SIC confusion matrix.

Class	C_1	C_2	C_3	C_4	C_5
C_1	**3/5**	0/5	0/5	0/5	0/5
C_2	0/5	**3/5**	0/5	0/5	0/5
C_3	0/5	0/5	**1/5**	0/5	4/5
C_4	0/5	0/5	0/5	**4/5**	0/5
C_5	0/5	0/5	0/5	0/5	**5/5**

Table 2.4: Dataset 2 PREDICT RIC confusion matrix.

2.5.3 Results Using the B-PREDICT Clustering Algorithm

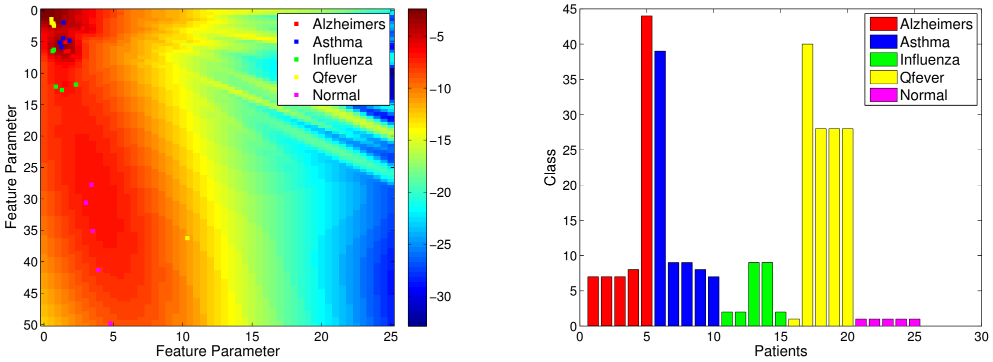
B-PREDICT was used to analyze Dataset 1 and 2 whose MLE fit Beta PDF feature space may be seen in Figures 2.6 and 2.7. For Dataset 1, $\alpha = 15$ and the error was set to 1, which gives a truncation of approximately $M = 73$, and 2,000 burn-in iterations were done in the blocked Gibbs sampler followed by 2,000 iterations for sampling. The confusion matrices with disease identified as C1-C6 are given in Tables 2.5 and 2.6. The true disease states are color coded for viewing ease, and the bar level for each measurement is the class membership per the final clustering. Using SIC, a rate of 70% was achieved as compared to a classification rate of 76.7% for RIC. For Dataset 2, the innovation parameter was set to $\alpha = 45$ and the error was 0.1, giving $M = 312$. The number of burn-in iterations was set to 2000 and the number of sample iterations was set to 3000. The SIC and RIC were each 64%.



(a) BPF features and DPGMM results.

(b) Clustering results.

Figure 2.6: B-PREDICT results for Dataset 1.



(a) BPF features and DPGMM results.

(b) Clustering results.

Figure 2.7: B-PREDICT results for Dataset 2.

Class	C_1	C_2	C_3	C_4	C_5	C_6
C_1	**5/5**	0/5	0/5	0/5	0/5	0/5
C_2	2/5	**2/5**	0/5	0/5	0/5	1/5
C_3	0/5	0/5	**4/5**	0/5	0/5	0/5
C_4	0/5	0/5	0/5	**3/5**	0/5	0/5
C_5	0/5	0/5	0/5	0/5	**5/5**	0/5
C_6	1/5	1/5	0/5	0/5	0/5	**2/5**

Table 2.5: Confusion matrix for Dataset 1 B-PREDICT using SIC.

Class	C_1	C_2	C_3	C_4	C_5	C_6
C_1	**5/5**	0/5	0/5	0/5	0/5	0/5
C_2	2/5	**2/5**	0/5	0/5	0/5	1/5
C_3	0/5	0/5	**4/5**	0/5	0/5	0/5
C_4	0/5	0/5	0/5	**5/5**	0/5	0/5
C_5	0/5	0/5	0/5	0/5	**5/5**	0/5
C_6	1/5	1/5	0/5	0/5	0/5	**2/5**

Table 2.6: Confusion matrix for Dataset 1 B-PREDICT using RIC.

Class	C_1	C_2	C_3	C_4	C_5
C_1	**3/5**	0/5	0/5	0/5	0/5
C_2	1/5	**2/5**	0/5	0/5	0/5
C_3	0/5	2/5	**3/5**	0/5	0/5
C_4	0/5	0/5	0/5	**3/5**	1/5
C_5	0/5	0/5	0/5	0/5	**5/5**

Table 2.7: Confusion matrix for Dataset 2 B-PREDICT using SIC.

Class	C_1	C_2	C_3	C_4	C_5
C_1	**3/5**	0/5	0/5	0/5	0/5
C_2	1/5	**2/5**	0/5	0/5	0/5
C_3	0/5	2/5	**3/5**	0/5	0/5
C_4	0/5	0/5	0/5	**3/5**	1/5
C_5	0/5	0/5	0/5	0/5	**5/5**

Table 2.8: Confusion matrix for Dataset 2 B-PREDICT using RIC.

2.6 Model challenges for PREDICT and B-PREDICT

Both PREDICT and B-PREDICT seek to encode information on the visible processes discernible to the viewer for immunosignaturing data. With PREDICT, the multidimensional PCA execution requires large, robust covariance matrices for correct PCA. While one shrinkage target is given in equation (2.5), there are multiple shrinkage targets for data with different structures, as given in [51]. As such, it is useful to have some prior knowledge of the data itself so that an appropriate shrinkage target may be selected. Furthermore, using PCA to decrease the overall dimensionality adds complexity if one is trying to determine which peptides play a role in each cluster. This is because now individual peptides are not represented by the dimensionality, but rather these are mapped into the PCA domain and are represented by combinations of peptides.

In the case of B-PREDICT, specifically dealing with the 2D BPF, while the drastic feature space reduction is especially favorable for complexity and time, relying solely on PDF shapes can eliminate significant underlying data patterns that would be useful for classification, especially those whose process is unobserved by the viewer. As this case uses MLE, it suffers from all drawbacks present in the MLE.

For example, MLE fitting using single mode PDFs will miss some cases with multimodal distributions, or cases were outliers corresponding to biological significance are present. This is especially true when assuming that there is no initial investigation by the analyzer into the PDF shape for each microarray, which is a reasonable assumption when dealing with data from many patients. In addition to these drawbacks, the DP-GMM has some limitations which affect both PREDICT and B-PREDICT. The first, and probably most important, limitation is that while cluster formation is potentially infinite, data membership to a given cluster is limited to one. This would mean that one patient may only be clustered in one group at any given time. When clusters correspond to disease types, this means that a patient is identified as having only one disease. Pathogenically it is possible for patients to have reactions to multiple diseases at any particular time, making the single membership requirement restrictive. In addition to this issue, the DPGMM only provides information on clustering membership, not on disease identification. In order to associate a particular disease with a particular cluster and provide patient diagnosis, a well defined training set would be needed, even though this training set is not required for clustering itself. Another drawback is that of the *a priori* assumptions and assignments. The DP-GMM requires an initial distribution assignment (in this case it is Gaussian) that is theoretically supposed to model the data well in the parameter space. There may be other distributions that provide better classification outcomes. This also leads to assumptions for the innovation parameter α. The *a priori* value of this innovation parameter is critical to cluster formation. Setting this value too large or to small may cause too strict or too loose of group membership associations, thus impacting the classification. Similarly, the truncation factor M is critical in that selecting a value too large may unnecessarily and negatively impact overall computation time. However, setting this parameter too small may result in very large truncation error.

Algorithm 1 Blocked Gibbs sampling for DP-GMM using an D-dimensional dataset $\mathbf{X}$

```
Input: Dataset $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, DP innovation parameter $\alpha$, Normal-Wishart hyperparameters $\boldsymbol{\mu}_{\boldsymbol{\mathcal{N}}}, \tau_{\mathcal{N}}, \xi_{\mathcal{W}}, \iota_{\mathcal{W}}$, DP truncation limit $M$.
Output: Samples $\{\boldsymbol{\mu}_m^{(i)}, \boldsymbol{\Sigma}_m^{(i)}, \mathbf{c}^{(i)}, \mathbf{w}^{(i)}\}_{i=1}^{L}$

Repeat for $i = 1, 2, \ldots,$ Gibbs iterations:
  1. Update for $\boldsymbol{\theta}_m^{(i)} = \{\boldsymbol{\mu}_m^{(i)}, \boldsymbol{\Sigma}_m^{-1(i)}\} \sim p\,(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m^{-1} | \mathbf{c}^{(i-1)}, \mathbf{X})$, $m = 1, \ldots, M$.
     (a) Let $\mathbf{X}_m = \{\mathbf{x}_n : c_n^{(i-1)} = m\}$ and $N_m = |\mathbf{X}_m|$, for $m = 1, \ldots, M$.
     (b) For all clusters, $m = 1, \ldots, M$, compute,
         $$\begin{aligned}
         \boldsymbol{\mu}_{\mathbf{x}_m} &= \frac{1}{N_m} \sum_{n:c_n^{(i-1)}=m} \mathbf{x}_n \\
         \boldsymbol{\Sigma}_{\mathbf{x}_m} &= \frac{1}{N_m} \sum_{n:c_n^{(i-1)}=m} (\mathbf{x}_m - \boldsymbol{\mu}_{\mathbf{x}_m})^2 \\
         \tilde{\boldsymbol{\mu}}_{\mathcal{N},m} &= \frac{\tau_{\mathcal{N}}\, \tilde{\boldsymbol{\mu}}_{\mathcal{N}} + N_m\, \boldsymbol{\mu}_{\mathbf{x}_m}}{\tau_{\mathcal{N}} + N_m}\,, \\
         \tilde{\tau}_{\mathcal{N},m} &= \tau_{\mathcal{N}} + N_m\,, \\
         \tilde{\iota}_{\mathcal{W},m} &= \iota_{\mathcal{W}} + \boldsymbol{\Sigma}_{\mathbf{x}_m} + \frac{\tau_{\mathcal{N}}\, N_m}{\tau_{\mathcal{N}} + N_m} (\mathbf{m} - \boldsymbol{\mu}_{\mathbf{x}_m}) (\mathbf{m} - \boldsymbol{\mu}_{\mathbf{x}_m})^T\,, \\
         \tilde{\xi}_{\mathcal{W},m} &= \xi_{\mathcal{W}} + N_m\,.
         \end{aligned}$$
     (c) Draw samples for $\boldsymbol{\Sigma}_m^{-1(i)}$ from the Wishart distribution, $\mathcal{W}\,(\boldsymbol{\Sigma}_m^{-1}; \tilde{\iota}_{\mathcal{W},m}, \tilde{\xi}_{\mathcal{W},m})$, for $m = 1, \ldots, M$.
     (d) Finally draw samples for $\boldsymbol{\mu}_m^{(i)}$ from the Normal distribution, $\mathcal{N}\,(\boldsymbol{\mu}_m; \tilde{\boldsymbol{\mu}}_{\mathcal{N},m}, \frac{\boldsymbol{\Sigma}_m^{(i)}}{\tilde{\tau}_{\mathcal{N},m}})$, for $m = 1, \ldots, M$.
  2. Update for $c_n^{(i)} \sim p\,(c_n | \boldsymbol{\mu}^{(i)}, \boldsymbol{\Sigma}^{-1(i)}, \mathbf{w}^{(i-1)}, \mathbf{X})$, $n = 1, \ldots, N$.
     (a) Let $q_{m,n} \triangleq w_m^{(i-1)} \mathcal{N}\,(\mathbf{x}_n; \boldsymbol{\mu}_m^{(i)}, \boldsymbol{\Sigma}_m^{(i)})$, $m = 1, \ldots, M$ and $n = 1, \ldots, N$.
     (b) Normalize $q'_{m,n} = \frac{q_{m,n}}{\sum_{m=1}^{M} q_{m,n}}$, $m = 1, \ldots, M$ and $n = 1, \ldots, N$.
     (c) Draw samples for $c_n^{(i)} \sim \sum_{m=1}^{M} q'_{m,n} \delta(c_n, m)$, $n = 1, \ldots, N$.
  3. Update for $w_m^{(i)} \sim p\,(w_m | \mathbf{c}^{(i)})$, $m = 1, \ldots, M$.
     (a) Draw samples $\beta_j \sim \mathrm{Beta}\Big(1 + N_m^*, \alpha + \sum_{m'=m+1}^{M} N_{m'}^*\Big)$, where $N_m^* \triangleq |\{n : c_n^{(i)} = m\}|$, $m = 1, \ldots, M$.
     (b) Finally evaluate $w_m^{(i)} = \beta_m \prod_{j=1}^{m-1} (1 - \beta_j)$, $m = 1, \ldots, M$.
```

Chapter 3

CLUSTERING USING BETA PROCESS FACTOR ANALYSIS

The beta process factor analysis (BPFA) model decomposes data into a linear combination of latent features for factors, providing information on the data's underlying structure. This is similar to the DP-GMM in that it also relies on a base distribution to describe the behavior of a parameter space. Unlike the DP, it is based on the beta process (BP) which is a true completely random measure [75, 84]. With mixture modeling and the DP-GMM, each element could only belong to a single group; this differs from BPFA where each item may have multiple group membership [53]. This may lead to the identification of underlying relationships between groups, offering a refined feature space for the clustering process. Additionally, while the DP-GMM requires a user specified feature or parameter space in addition to the underlying distributions as model parameters, BPFA only requires the specification of the underlying model distributions and learns the number of underlying latent features.

3.1 Beta process theory and related representations

The beta process [85] is useful in Bayesian nonparametric modeling for latent features [86–88], especially given the implementable conjugate prior relationship inherent in the prior. A generalization was presented in [89] where the BP was shown to be a special case of the kernel beta process. The construction of the beta process may be described as [86, 90]:

$$H(B_k) \sim Beta(\alpha_B H_0(B_k), \alpha_B(1 - H_0(B_k)) \tag{3.1}$$

where $k = 1, ..., K$ are the individual partitions or latent features, B_k is the partition itself in the space, and α_B is a positive scalar. H_0 is the continuous probability measure on the same measurable space. Thus, as $K \to \infty$, $H_k \to H$. Similarly, the BP has a stick-breaking representation [90, 91]:

$$\theta_{ij} \overset{\text{iid}}{\sim} H_0/\gamma, \tag{3.2}$$

$$V_{ij} \overset{\text{iid}}{\sim} \text{Beta}(1, \alpha_B), \tag{3.3}$$

$$H(\boldsymbol{\theta}) = \sum_{i=1}^{\infty} \sum_{j=1}^{C_i} \prod_{l=1}^{i-1} (1 - V_{ij}) \delta_{\theta_{ij}}, \tag{3.4}$$

$$C_i \overset{\text{iid}}{\sim} \text{Poisson}(\gamma) \tag{3.5}$$

where $\gamma = H_0(\{B_1...B_K\})$ and *iid* stands for independent and identically distributed.

While the BP is infinite in feature number, a practical truncation limitation is often applied to approximate the BP. As the truncation limit increases, it will become a closer approximation of the theoretical infinite case. A finite approximation and a two parameter version may be generated [86]:

$$H_K = \sum_{k=1}^{K} (\pi_k \delta_{\phi_k}), \tag{3.6a}$$

$$\phi_k \overset{\text{iid}}{\sim} H_0, \tag{3.6b}$$

$$\pi_k \sim \text{Beta}(a/K, b(K-1)/K), \; k = 1 \ldots K \tag{3.6c}$$

It should be noted that $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_K\}$ serves as a new measure of the space, and it is a parameter for a Bernoulii process, $z_{k,n}$ which will serve as a binary indicator variable for the latent features to be discovered [86, 92]:

$$z_{k,n} \sim Bernoulli(\pi_k) \tag{3.7}$$

The combination of Equations (3.6) and (3.7) form a prior for the BPFA. Another similarly related process is the Indian buffet process (IBP). The structure is the same as in Equations (3.6) and (3.7), but the second parameter of the beta distribution, $b(K-1)/K$ simply becomes equal to 1 [92]. While K is indicative of the truncation coefficient, it is also the maximum number of latent features in BPFA. As such, it is important for K to be sufficiently large to fully capture the latent feature interactions within a data set. In practicality, K should not be chosen so large as to impede quick calculation and convergence for the parameter estimation. The K value selection is discussed in Section 3.3.

3.2 Beta process factor analysis and blocked Gibbs sampler

As in the case of the DP-GMM, a blocked Gibbs sampler is employed for the BPFA model parameter estimation. Both a Gibbs sampler method and a variational Bayes (VB) method can be used to implement BP algorithms [93, 94]. While it is noted that the Gibbs sampling method may require more iterations to converge than a VB method [93], its ease of implementation makes it a suitable selection for immunosignaturing work. Further extensions using collapsed and accelerated Gibbs sampling with regards to the IBP are discussed in [95]. An MCMC method was selected for its overall flexibility. The blocked Gibbs sampler was chosen in order to improve overall performance. The blocked Gibbs sampler also relies heavily on conjugate priors. In order to construct the algorithm, it is necessary to devise a model to describe the data that is capable of exploiting these conjugate prior relationships. Using [53, 86], this BPFA model is described as:

$$\mathbf{X} = \mathbf{\Phi}\mathbf{Z} + \mathbf{E} \tag{3.8}$$

where $\mathbf{X}$ is the immunosignaturing data, matrix $\mathbf{\Phi}$ and binary matrix $\mathbf{Z}$ are parameter matrices that describe latent features, and $\mathbf{E}$ is the error. For a single dataset (i.e., patient in the case of immunosignaturing), this can be expressed as:

$$\mathbf{x}_n = \sum_{k=1}^{K} z_{k,n}\boldsymbol{\phi}_k + \boldsymbol{e} \tag{3.9}$$

where $\mathbf{X} = [\mathbf{x}_1^{\mathsf{T}} \ \ldots \ \mathbf{x}_N^{\mathsf{T}}]$ is the $(D \times N)$ data matrix, $\mathbf{\Phi} = [\boldsymbol{\phi}_1^{\mathsf{T}} \ \ldots \ \boldsymbol{\phi}_K^{\mathsf{T}}]$ is the $(D \times K)$ latent factor matrix, $\mathbf{Z} = [\mathbf{z}_1^{\mathsf{T}} \ \ldots \ \mathbf{z}_N^{\mathsf{T}}]$ is the $(K \times N)$ factor weight binary matrix, and $\mathbf{z}_n$ is the $(1 \times K)$ binary vectors of the nth patient.

A weighted version of this decomposition can be used depending on the required factor model output [86]. For further discussion, $\mathbf{X}$ is considered to be DxN with D being the data dimensionality and N being the number data points in each dimensionality. Additionally, $\mathbf{\Phi}$ is DxK where K is the truncation value or maximum number of latent features. Finally, $\mathbf{Z}$ is KxN and $\mathbf{E}$ is DxN. The additional intermediate random variable $\boldsymbol{\pi}$ is used as a precursor for describing $\mathbf{Z}$ per equation (3.7). For this model, $\mathbf{Z}$ is considered to be Bernoullli distributed, $\boldsymbol{\pi}$ is Beta distributed, $\mathbf{\Phi}$ is Gaussian, and the variance of the error $\mathbf{E}$ is Inverse-Gamma. This makes $\mathbf{X}$ also Gaussian with a mean of $\mathbf{\Phi Z}$ and a variance of $\sigma_e^2\mathbf{I}$ where $\mathbf{I}$ is the identity matrix. The model parameter relationships are described as [86]:

$$\mathbf{X} \sim \mathcal{N}(\mathbf{\Phi Z}, \sigma_e^2\mathbf{I}), \tag{3.10a}$$

$$\sigma_e^2 \quad \mathcal{I}\Gamma(c, d), \tag{3.10b}$$

$$\boldsymbol{\phi}_k \sim \mathcal{N}(0, \mathbf{\Sigma}_{\phi_k}), \tag{3.10c}$$

$$z_{k,n} \sim \mathcal{B}ernoulli(\pi_k), \tag{3.10d}$$

$$\pi_k \sim Beta(\alpha, \beta) \tag{3.10e}$$

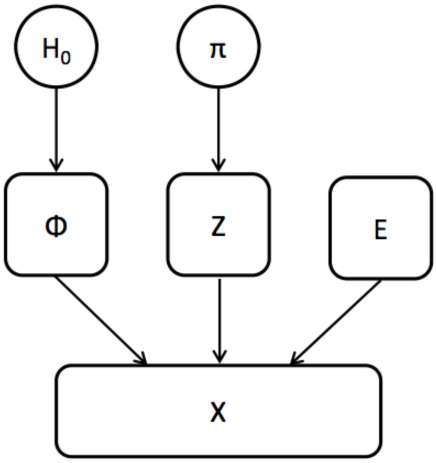


Figure 3.1: BP block diagram.

where n refers to the nth element of N data points, $\boldsymbol{\phi_k}$ refers to the kth column of $\mathbf{\Phi}$, and $z_{k,n}$ refers to the nth element of the kth row of the $\mathbf{Z}$ matrix. The relationship between these random variables is given in Figure 3.1.

The conjugate prior relationships are the basis for the MCMC technique considered here for BPFA. When this is applied to the blocked Gibbs sampler framework for each of the BPFA parameters for the ith iteration, this becomes:

$$\begin{aligned}
\pi_k^{(i)} &\sim p(\pi|\mathbf{Z}^{(i-1)}, \alpha_0^{(i)}, \beta_0^{(i)}),\ k = 1 \dots K, && \text{(3.11a)}\\
\boldsymbol{\phi}_k^{(i)} &\sim p(\boldsymbol{\phi}_k|\Phi^{(i-1)}, \mathbf{Z}^{(i-1)}, \sigma_e^{2(i-1)}, \mathbf{X}),\ k = 1 \dots K, && \text{(3.11b)}\\
z_{k,n}^{(i)} &\sim p(z_{i,k}|\mathbf{\Phi}^{(i)}, \mathbf{Z}^{(i-1)}, \sigma_e^{2(i-1)}, \pi^{(i)}, \mathbf{X}),\ k = 1 \dots K,\ n = 1 \dots N, && \text{(3.11c)}\\
\sigma_e^{2(i)} &\sim p(\sigma_e^2|\mathbf{\Phi}^{(i)}, \mathbf{Z}^{(i)}, \sigma_e^{2(i-1)}, \mathbf{X}) && \text{(3.11d)}
\end{aligned}$$

There are also hyperparameters (a_z, b_z, c_z, and d_z) that are also updated in the blocked Gibbs sampler, but they utilize different priors and are conditional functions of both the initialized value for each hyper-parameter, as well as the other estimated matrices in each iteration. These hyper-parameters are used to describe other distributions and are only indirectly related to the estimated model parameters of interest. It should be noted that the update order within the blocked Gibbs sampler does not matter as long as the most recent updates are used for each new estimation. The blocked Gibbs sampler algorithm with specific update equations for BPFA is given in Algorithm 2.

Noting that $||\cdot||$ is the norm of the expression, combined together, this provides the following conditional PDFs:
β-distributed $\pi_k^{(i)}$,

$$\beta\big(\pi_k^{(i)}; a_z^{(i)}, b_z^{(i)} \bigm| \mathbf{Z}^{(i-1)}, a, b\big) \tag{3.12}$$

$$a_z^{(i)} = \frac{a}{K} + \sum_{n=1}^{N} z_{k,n}^{(i-1)} \tag{3.13}$$

$$b_z^{(i)} = \frac{b(K-1)}{K} + N - \sum_{n=1}^{N} z_{k,n}^{(i-1)}\,, \tag{3.14}$$

Gaussian distributed $\boldsymbol{\phi}_k^{(i)}$,

$$g\big(\boldsymbol{\phi}_k^{(i)}; \boldsymbol{\mu}_{\boldsymbol{\phi}}^{(i)}, \tilde{\boldsymbol{\Sigma}}_{\boldsymbol{\phi}}^{(i)} \bigm| \boldsymbol{\Phi}^{(i-1)}, \mathbf{Z}^{(i-1)}, \sigma_e^{2\,(i-1)}, \mathbf{X}\big) \tag{3.15}$$

$$\tilde{\boldsymbol{\Sigma}}_{\boldsymbol{\phi}}^{(i)} = \Big(\boldsymbol{z}_k^{(i-1)} \boldsymbol{z}_k^{\mathtt{T},(i-1)} \mathbf{I}_D\, {\sigma_e}^{-2\,(i-1)} + \boldsymbol{\Sigma}_{\phi}^{-1}\Big)^{-1} \tag{3.16}$$

$$\boldsymbol{\mu}_{\boldsymbol{\phi}}^{(i)} = \Big[\Big(\boldsymbol{z}_k^{(i-1)} \boldsymbol{z}_k^{\mathtt{T}\,(i-1)} \mathbf{I}_D {\sigma_e}^{-2\,(i-1)} + \boldsymbol{\Sigma}_{\phi}^{-1}\Big)^{-1} \\ {\sigma_e}^{-2\,(i-1)} \sum_{n=1}^{N} z_{k,n}^{(i-1)} \sum_{\substack{k'=1 \\ k'\neq k}}^{K} z_{k',n}^{(i-1)} \boldsymbol{\phi}_{k'}^{\mathtt{T}\,(i-1)}\Big]\,, \tag{3.17}$$

where $\mathbf{z}_k$ is the kth row of $\mathbf{Z}$ and Bernoulli distributed $z_{k,n}^{(i)}$

$$\text{Bernoulli}\big(z_{k,n}^{(i)}; \psi^{(i)} \big| \boldsymbol{\Phi}^{(i)}, \mathbf{Z}^{(i-1)}, \sigma_e^{2\,(i-1)}, \boldsymbol{\pi}^{(i)}, \mathbf{X}\big) \tag{3.18}$$

$$\psi^{(i)} = \left[1 + \frac{1-\pi_k^{(i)}}{\pi_k^{(i)}} \exp\left(\frac{(\sigma_e^2)^{(i-1)}}{2} \left\| \mathbf{x}_n - \sum_{\substack{k'=1 \\ k'\neq k}}^{K} z_{k',n}^{(i-1)} \boldsymbol{\phi}_{k'}^{(i)} \right\|^2 - \right.\right.$$

$$\left.\left. \left\| \mathbf{x}_n - \boldsymbol{\phi}_k^{(i)} - \sum_{\substack{k'=1 \\ k'\neq k}}^{K} z_{k',n}^{(i-1)} \boldsymbol{\phi}_{k'}^{(i)} \right\|^2 \right) \right]^{-1}, \tag{3.19}$$

and inverse Gamma distributed $\sigma_e^{2\,(i)}$,

$$\mathcal{I}\Gamma\left(\sigma_e^{2\,(i)}; c_z^{(i)}, d_z^{(i)} \;\middle|\; \boldsymbol{\Phi}^{(i)}, \mathbf{Z}^{(i)}, \mathbf{X} \right) \tag{3.20}$$

$$c_z^{(i)} = c + (ND/2) \text{ and} \tag{3.21}$$

$$d_z^{(i)} = d + \frac{1}{2} \sum_{n=1}^{N} \left\| \mathbf{x}_n - \sum_{k'=1}^{K} z_{k',n}^{(i)} \boldsymbol{\phi}_{k'}^{(i)} \right\|^2 . \tag{3.22}$$

3.3 Selection of K for BPFA

The remaining item to be defined for BPFA for the immunosignaturing model is the selection of the user defined truncation parameter K. When using this model and referring to $\boldsymbol{\pi}$ in both Equation (3.10) and (3.6), α_0 and β_0 are defined as [86]:

$$\alpha_0 = \frac{a}{K}, \tag{3.23a}$$

$$\beta_0 = \frac{b(K-1)}{K} \tag{3.23b}$$

where a and b are similar to the innovation parameter α in the DP-GMM. In the BPFA, these parameters help to define the multi-feature presence of each item to

Algorithm 2 Blocked Gibbs sampling for BPFA using an D-dimensional dataset $\mathbf{X}$

```
Input: Dataset $\mathbf{X} = [\mathbf{x_1}, \ldots, \mathbf{x_N}]$, beta process positive scalars $a$ and $b$, the truncation limit $K$, and inverse gamma hyperparameters $c$ and $d$.
Output: Samples $\{\boldsymbol{\pi}, \boldsymbol{\Phi}, \mathbf{Z}, \boldsymbol{\sigma}_e^2\}_{j=1}^{L}$ from the posterior pdf $P(\boldsymbol{\pi}, \boldsymbol{\Phi}, \mathbf{Z}, \boldsymbol{\sigma}_e^2 | \mathbf{X})$.

Repeat for $j = 1, 2, \ldots,$ Gibbs iterations:
    1. Update for $\{\pi_k\} \sim \beta\big(\pi_k^{(i)}; a_z^{(i)}, b_z^{(i)} \mid \mathbf{Z}^{(i-1)}, a, b\big),\ k = 1 \ldots K$
        (a) Draw: $\beta\big(\pi_k^{(i)}; \frac{a}{K} + \sum_{n=1}^{N} z_{k,n}^{(i-1)}, \frac{b(K-1)}{K} + N - \sum_{n=1}^{N} z_{k,n}^{(i-1)} \big| \mathbf{Z}^{(i-1)}, a, b\big)$
    2. Update for $\{\mathbf{z}_k\} \sim \mathrm{Be}\big(z_{k,n}^{(i)}; \psi^{(i)} \big| \boldsymbol{\Phi}^{(i-1)}, \mathbf{Z}^{(i-1)}, \sigma_e^{2\,(i-1)}, \boldsymbol{\pi}^{(i)}, \mathbf{X}\big),\ k = 1 \ldots K$ .
        (a) Draw: $\mathrm{Be}\left(z_{k,n}^{(i)}; \left[1 + \frac{1-\pi_k^{(i)}}{\pi_k^{(i)}} \exp\left(\frac{(\sigma_e^2)^{(i-1)}}{2} \left\| \mathbf{x}_n - \sum_{\substack{k'=1\\k'\neq k}}^{K} z_{k',n}^{(i-1)} \boldsymbol{\phi}_{k'}^{(i-1)} \right\|^2 - \left\| \mathbf{x}_n - \boldsymbol{\phi}_k^{(i-1)} - \sum_{\substack{k'=1\\k'\neq k}}^{K} z_{k',n}^{(i-1)} \boldsymbol{\phi}_{k'}^{(i-1)} \right\|^2 \right)\right]^{-1} \middle| \boldsymbol{\Phi}^{(i-1)}, \mathbf{Z}^{(i-1)}, \sigma_e^{2\,(i-1)}, \boldsymbol{\pi}^{(i)}, \mathbf{X}\right)$
    3. Update for $\{\boldsymbol{\Phi}\} \sim g\big(\boldsymbol{\phi}_k^{(i)}; \boldsymbol{\mu}_{\boldsymbol{\phi}}^{(i)}, \tilde{\boldsymbol{\Sigma}}_{\boldsymbol{\phi}}^{(i)} \mid \boldsymbol{\Phi}^{(i-1)}, \mathbf{Z}^{(i)}, \sigma_e^{2\,(i-1)}, \mathbf{X}\big)$, k=1 ... K
        (a) Draw: $g\big(\boldsymbol{\phi}_k^{(i)}; \boldsymbol{\mu}_{\boldsymbol{\phi}}^{(i)}, \tilde{\boldsymbol{\Sigma}}_{\boldsymbol{\phi}}^{(i)} \big| \boldsymbol{\Phi}^{(i-1)}, \mathbf{Z}^{(i)}, \sigma_e^{2\,(i-1)}, \mathbf{X}\big)$.
        (b) Define: $\tilde{\boldsymbol{\Sigma}}_{\boldsymbol{\phi}}^{(i)} = \left(\boldsymbol{z}_k^{(i)} \boldsymbol{z}_k^{\mathtt{T},(i)} \mathbf{I}_D\, \sigma_e^{-2\,(i-1)} + \boldsymbol{\Sigma}_\phi^{-1}\right)^{-1}$
        (c) Define: $\boldsymbol{\mu}_{\boldsymbol{\phi}}^{(i)} = \left[\tilde{\boldsymbol{\Sigma}}_{\boldsymbol{\phi}}^{(i)} \sigma_e^{-2\,(i-1)} \sum_{n=1}^{N} z_{k,n}^{(i)} \sum_{\substack{k'=1\\k'\neq k}}^{K} z_{k',n}^{(i)} \boldsymbol{\phi}_{k'}^{\mathtt{T}\,(i-1)}\right]$
    4. Update for $\{\boldsymbol{\sigma}_e^2\} \sim \mathcal{I}\Gamma\left(\sigma_e^{2\,(i)}; c_z^{(i)}, d_z^{(i)} \,\middle|\, \boldsymbol{\Phi}^{(i)}, \mathbf{Z}^{(i)}, \mathbf{X}\right)$
        (a) Draw: $\mathcal{I}\Gamma\left(\sigma_e^{2\,(i)}; c_z^{(i)}, d_z^{(i)} \,\middle|\, \boldsymbol{\Phi}^{(i)}, \mathbf{Z}^{(i)}, \mathbf{X}\right)$
        (b) Define: $c_z^{(i)} = c + (ND/2)$
        (c) Define: $d_z^{(i)} = d + \frac{1}{2} \sum_{n=1}^{N} \left\| \mathbf{x}_n - \sum_{k'=1}^{K} z_{k',n}^{(i)} \boldsymbol{\phi}_{k'}^{(i)} \right\|^2$
```

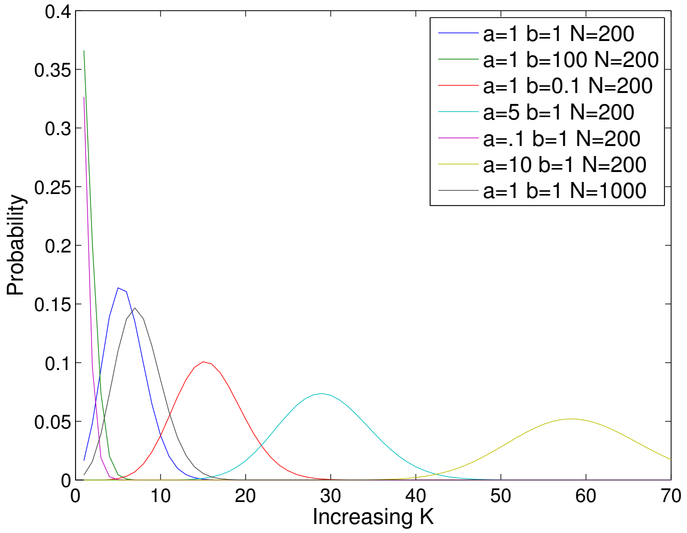


Figure 3.2: Example distributions for K truncation selection.

be classified. Thus, while K is user defined, there is an interrelationship between a, b, and K that affects the latent feature groupings during estimation. Furthermore, truncation of K far below the true underlying feature amount will result in sub-optimal latent feature memberships, further affecting the classification. As such, it is desired to know approximately the value of K. Further complicating the selection is the interaction of N, which is defined by the input data, on the Poisson distribution indicated in Equation (3.2). This interaction is given as [86]:

$$\lambda_P = \sum_{n=1}^{N} \frac{a}{b+n-1} \tag{3.24}$$

where λ_P is the parameter for the Poisson distribution. An example of this effect is given in Figure 3.2. Thus, one should select a value of K that encompasses enough of the Poisson distribution specified by the parameter given in Equation (3.24).

The IBP has very similar functionality to the BPFA algorithm in that a beta process prior is used as in equation (3.6c), but the innovation parameters are changed such that $\alpha_0 = a/K$ but $\beta_0 = 1$ [92]. The link between the IBP and the beta process may be thought of as CRP:DP::IBP:BP [96]. Further discussion of the IBP may be found in [97].

3.4 BPFA and immunosignaturing input data format

When dealing with immunosignaturing data, there are at least two possible use cases for analysis of incoming patient data. While each case does not impact how the BPFA is modeled or executed, it does impact how the input data is used within the BPFA. In each case, it will effectively change the random variable that is used to further classify patients based on underlying disease state. In this work, reduction in the number of peptides for analysis is performed by using either PCA or BPF as discussed in Sections 2.1 and 2.2. Further refinement is then done by the BPFA.

In the first format, the feature model still follows the format of Equations (3.8) and (3.10), but the dimension D is defined as the number of patients while the number of data points N is defined as the microarray intensity measurements, or some dimensionality reduction thereof, for each patient. This case would be useful when additional microarray data points are received for patients over time, but the overall number of patients does not change. This then means that the estimated value of the $\mathbf{\Phi}$ matrix, represented as $\hat{\mathbf{\Phi}}$, and found after using the blocked Gibbs sampler, would be the parameter that describes the relationship between patients and underlying features. It is these latent features that are thought to represent the underlying states (i.e. diseases) for patients. In this case, the estimated $\mathbf{Z}$ matrix, hereafter represented as $\hat{\mathbf{Z}}$, would represent the relationship between peptides or combinations of peptides (after dimensionality reduction) to latent features.

In the second format, which also follows the general model structure given in Equations (3.8) and (3.10), dimensionality D is defined as the number of peptide sequences, or peptide sequence combinations after dimensionality reduction. This then means that N is defined as the number of patients being analyzed. For this case, the $\hat{\mathbf{\Phi}}$ matrix from the blocked Gibbs sampler iterations will illustrate the relationship between latent features and peptides (or peptide combinations after dimensionality reduction). This may be used to show which peptide or peptide combinations play a role in various disease states, similar to how the $\hat{\mathbf{Z}}$ matrix was used in the previous format. This then means that the $\hat{\mathbf{Z}}$ matrix derived from the execution of the blocked Gibbs sampler may be use to highlight the underlying feature representations for each patient, and thus the underlying disease state for each patient. This is similar to how the $\hat{\mathbf{\Phi}}$ matrix was used in the previous format.

3.5 Clustering Using BPFA

3.5.1 BPFA with Z-matrix and PCA Features

Once the $\hat{\mathbf{\Phi}}$ and $\hat{\mathbf{Z}}$ matrices are found from BPFA, it is possible to use a variety of classification or clustering schemes to determine patient groupings based on diseases. The first is the Z-PREDICT or **Z**-matrix from **P**CA **RE**solution and **D**iscovery for **I**mmunosignature **C**lassification **T**esting, uses the log of the PCA feature space as first presented in PREDICT as the input to the BFPA. Since the $\hat{\mathbf{Z}}$ matrix is a binary feature matrix, features are either present, as represented by a 1, or not present, as represented by a 0.

If there are not too many latent features found during BPFA application, then it may be possible to simply inspect the feature combinations and determine which patients contain similar feature representations. This can be time consuming as the data grows, and therefore a more formal representation is desired. However, the $\hat{\mathbf{Z}}$ matrix then needs to be modified such that incorrect comparisons can be penalized without negatively impacting correct correlation matches. This is accomplished by replacing all zero entries with a negative one, and is known then as the matrix $\mathbf{Y}$. The replacement of these strategic values introduces the concept of penalty and reward criteria for the $\hat{\mathbf{Z}}$ matrix which can be modified depending on the clustering problem. This matrix is then multiplied by the transpose of itself to find the feature vector correlations amongst all vectors. This can mathematically be described as the following modifications to the $\mathbf{Z}$ matrix:

$$\mathbf{Y} = (2\mathbf{Z} - \mathbf{1}), \tag{3.25a}$$

$$\mathbf{C} = \mathbf{Y}^T\mathbf{Y} \tag{3.25b}$$

This may also be thought of as taking the non-normalized dot product of each K dimensional binary feature vector with every other binary feature vector. Individual comparisons with high $\mathbf{C}$ values are more likely to be classified together correctly. For example, consider the $\hat{\mathbf{Z}}$ matrix given in Figure 3.3. This matrix is presented in left ordered form (LOF) [92] and is transposed for readability with features being all zeros removed. Each patient has an associated set of features that may be represented by a binary feature vector.

For example, consider patient 1 whose binary feature vector is $\mathbf{z_1} = [1, 1, 0, 0, 0, 0]$. After equation (3.25a), this becomes $\mathbf{y_1} = [1, 1, -1, -1, -1, -1]$. In order to see if this patient should be clustered with patient 2, whose modified binary feature vector is $\mathbf{y_2} = [1, 1, -1, -1, -1, -1]$, find the correlation (or dot product) of these two vectors. The result of this correlation is $\mathbf{y_1} \cdot \mathbf{y_2} = 6$. Now, compare the modified binary feature vector for patient 1 with that for patient 25 ($\mathbf{y_{25}} = [-1, 1, -1, -1, -1, 1]$); the correlation in this case is four. Since the correlation between patient 1 and patient 25 is lower than the correlation between patient 1 and patient 2, it is more likely

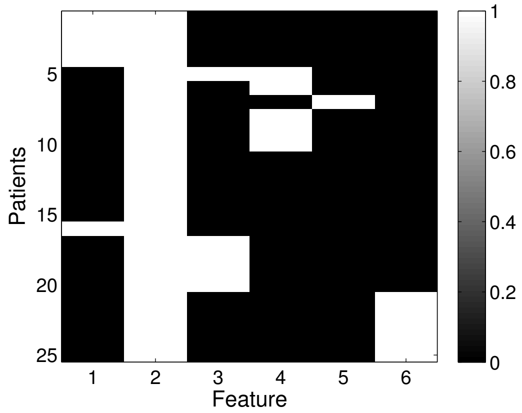


Figure 3.3: Example $\hat{\mathbf{Z}}$ matrix

that patient 1 and patient 2 should be clustered together than patient 1 and patient 25. This pairwise comparison may be done for all the patients and a ranking of their correlations may be achieved.

3.5.2 BPFA with Z-matrix and Beta PDF Fitting Features

In Z-PREDICT, PCA was used for dimensionality reduction. Similarly, BPF may be used as well as described in section 2.2. In the ZB-PREDICT method (**Z**-matrix from **B**PDF **RE**solution and **D**iscovery for **I**mmunosignature **C**lassification **T**esting), scaled BPF is used to reduce the immunosignature data dimensionality. This is then fed into the BPFA for latent feature discovery. The latent feature binary $\hat{\mathbf{Z}}$ matrix is then modified according to equation (3.25a). Again, the higher the $\mathbf{C}$ value, the more likely two patients are to be grouped together due to the fact that their latent feature spaces are closer matches to one another, as is seen when using the $\mathbf{Z}$ matrix modifications to penalize incorrect matches.

3.5.3 BPFA with Φ Matrix and PCA

Another method, Φ-PREDICT, may be used to determine patient clustering by disease as well. This is also known as **Φ P**CA **RE**solution and **D**iscovery for **I**mmunosignature **C**lassification **T**esting. As previously discussed, this requires the input data matrix to be transposed from that discussed in Z-PREDICT or ZB-PREDICT. As opposed to previous methods, $\hat{\mathbf{\Phi}}$ (the MCMC estimate of $\mathbf{\Phi}$) is a non-binary feature matrix governed by the distribution given in equation (3.10). Due to the non-binary nature of this matrix, it may be more difficult to determine patient groups by simple inspection. As such, a simple quantization scheme may be employed to determine patient groups. Each latent feature may be assigned to a pre-determined user specified quantization level. This will lead to a quantized version of $\hat{\mathbf{\Phi}}$, which means that each patient will have different quantized latent feature representations. By inspection, it will be possible to see which patients contain feature combinations at identical quantization levels and group these patients together. It should be noted that the results from the $\hat{\Phi}$ matrix may be fed into the DP-GMM for further adaptive classification. This was not investigated in this work due to other positive results using alternative methods.

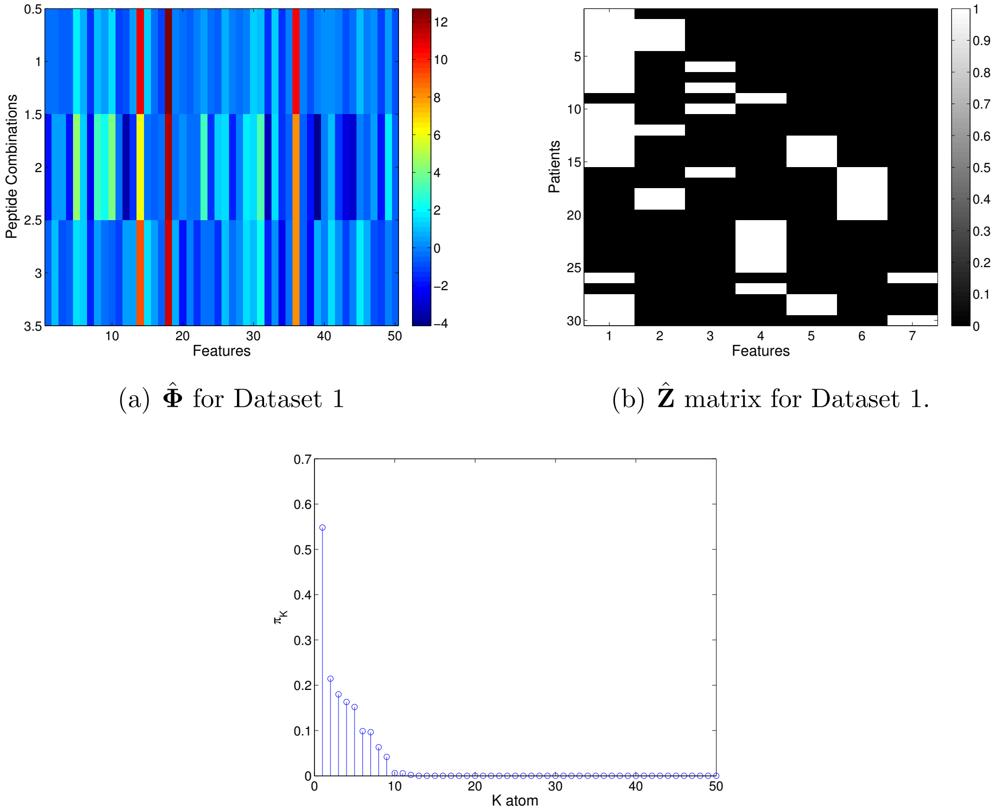
It should be noted that there is still an RIC and SIC application for these classification results. However, since the Φ-PREDICT method is much more user involved in terms of specifying clustering parameters, the definition of SIC and RIC are slightly changed. SIC is still the exact match of all features but now is extended to all quantization levels. The definition of RIC is more user specific. For RIC, it is still the matching of features, but in this case the user may elect to allow for some variation amongst features that do not match. This can be though of as allowing the majority of features to be correctly matched (example results in [44], which were produced before the inception of RIC and SIC), or it may be thought of as allowing feature variance within a certain number of quantization levels. Additionally, as the number of examined features increases, the less likely they are to match at every quantization level all the time, making the analysis overly restrictive.

3.6 BPFA Clustering Results

3.6.1 Results of BPFA with PCA Features

In this case, the input data matrix to the BPFA will be transposed. This means that D is the number of peptide combinations selected from the PCA of the median intensity data, and N is the number of patients. For Dataset 1 the resulting $\hat{\mathbf{\Phi}}$ and $\hat{\mathbf{Z}}$ matrices as well as the estimated reordered $\hat{\pi_k}$ values are presented in Figure 3.4. The $\hat{\mathbf{Z}}$ matrix is in the LOF configuration [92] and transposed. Features containing no entries were removed in order to easily show the feature relationships. In this case, it is the $\hat{\mathbf{\Phi}}$ matrix that illustrates the relationship between the peptide combinations and underlying features and $\hat{\mathbf{Z}}$ illustrates the relationship between features and patients. Three PCA components were used in this analysis which represents 93.5% of the data and $K = 50$. A total of 2000 burn-in iterations and 3000 sample iterations were used for the blocked Gibbs sampler. The confusion matrices for SIC and RIC cases are given in Tables 3.1 and 3.2. The SIC result was 60% and the RIC result was 73.3%.

For Dataset 2 the resulting $\hat{\mathbf{\Phi}}$, $\hat{\mathbf{Z}}$, $\hat{\pi_k}$ are presented in Figure 3.5. Once again, $\hat{\mathbf{Z}}$ is presented in LOF configuration, transposed, and with features have zero entries removed. Five PCA components were used for this analysis which accounts for 93.5% of the data and $K = 50$ for this analysis. A total of 3000 burn-in iterations and 3000 sample iterations were used. The resulting SIC and RIC were both 88%.

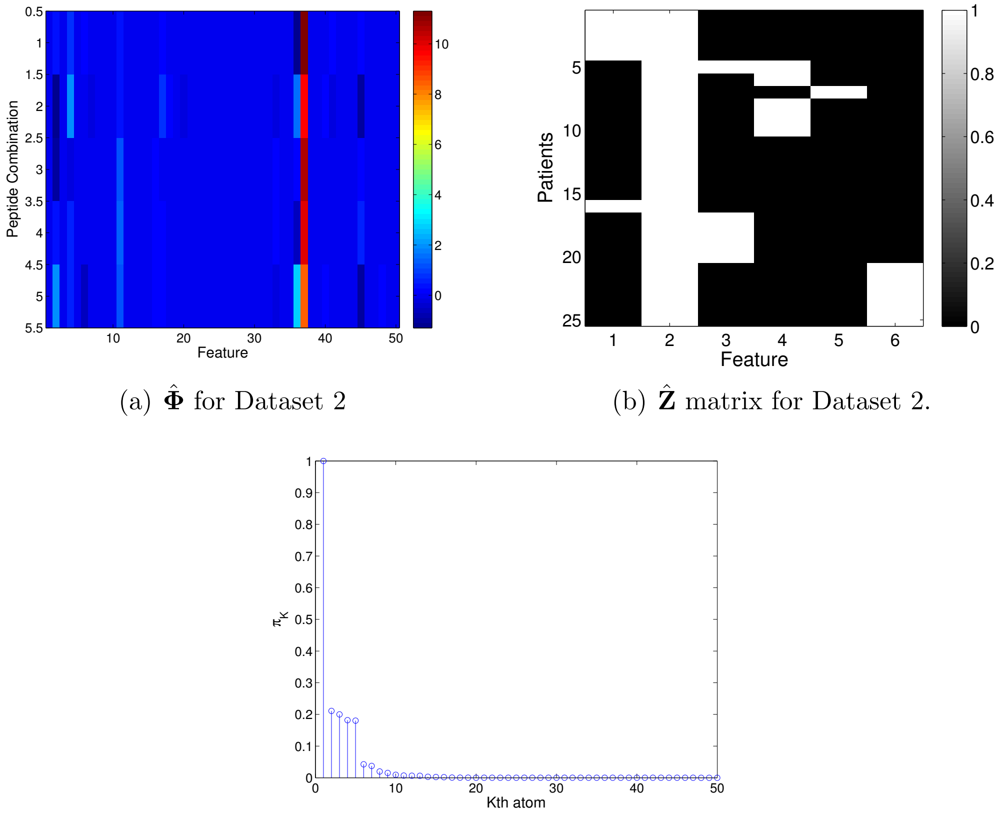


(a) $\hat{\mathbf{\Phi}}$ for Dataset 1

(b) $\hat{\mathbf{Z}}$ matrix for Dataset 1.

(c) $\hat{\pi_k}$ values in descending order

Figure 3.4: BPFA results from Dataset 1

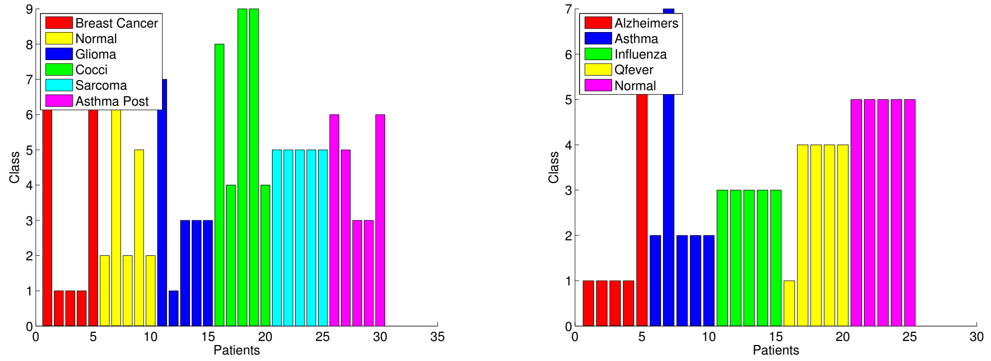


(a) $\hat{\mathbf{\Phi}}$ for Dataset 2

(b) $\hat{\mathbf{Z}}$ matrix for Dataset 2.

(c) $\hat{\pi}_k$ values in descending order

Figure 3.5: BPFA for Dataset 2



(a) Dataset 1 results.

(b) Dataset 2 results.

Figure 3.6: Z-PREDICT classification results.

Class	C_1	C_2	C_3	C_4	C_5	C_6
C_1	**3/5**	0/5	0/5	0/5	0/5	0/5
C_2	0/5	**3/5**	0/5	0/5	1/5	0/5
C_3	1/5	0/5	**3/5**	0/5	0/5	0/5
C_4	0/5	0/5	0/5	**2/5**	0/5	0/5
C_5	0/5	0/5	0/5	0/5	**5/5**	0/5
C_6	0/5	0/5	2/5	0/5	1/5	**2/5**

Table 3.1: Dataset 1 confusion matrix for Z-PREDICT using SIC.

Class	C_1	C_2	C_3	C_4	C_5	C_6
C_1	**5/5**	0/5	0/5	0/5	0/5	0/5
C_2	0/5	**3/5**	0/5	0/5	1/5	0/5
C_3	1/5	0/5	**3/5**	0/5	0/5	0/5
C_4	0/5	0/5	0/5	**4/5**	0/5	0/5
C_5	0/5	0/5	0/5	0/5	**5/5**	0/5
C_6	0/5	0/5	2/5	0/5	1/5	**2/5**

Table 3.2: Dataset 1 confusion matrix for Z-PREDICT using RIC.

Class	C_1	C_2	C_3	C_4	C_5
C_1	**4/5**	0/5	0/5	0/5	0/5
C_2	0/5	**4/5**	0/5	0/5	0/5
C_3	0/5	0/5	**5/5**	0/5	0/5
C_4	1/5	0/5	0/5	**4/5**	0/5
C_5	0/5	0/5	0/5	0/5	**5/5**

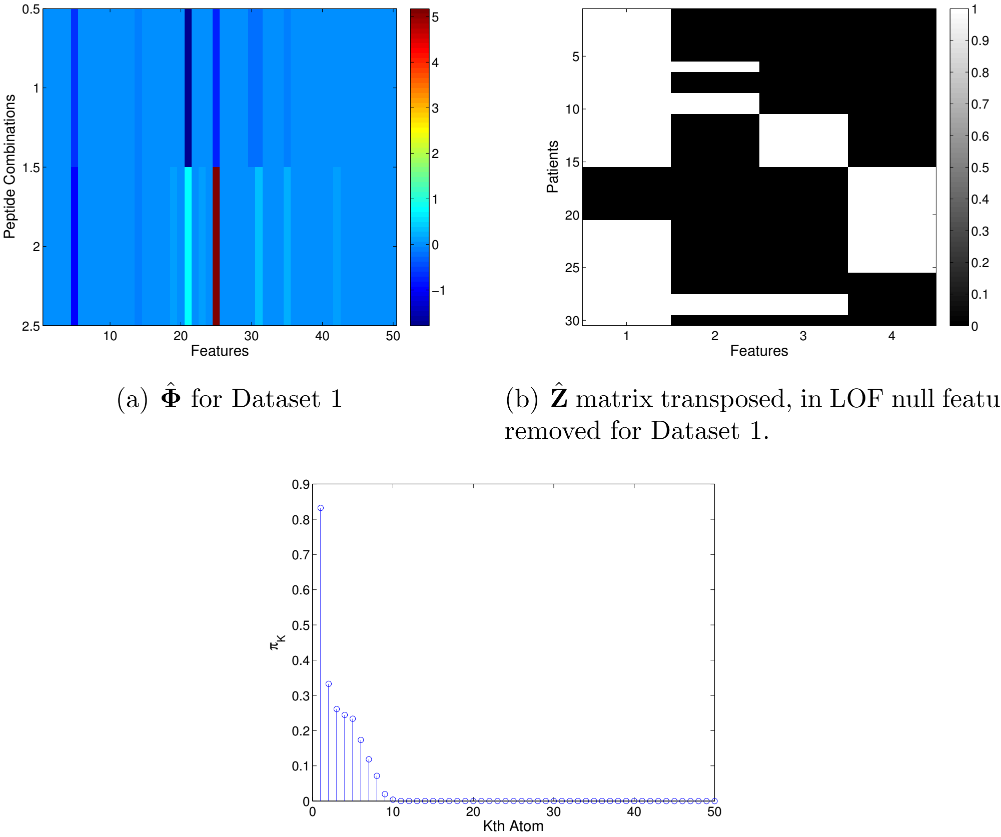
Table 3.3: Dataset 2 confusion matrix for Z-PREDICT using SIC.

Class	C_1	C_2	C_3	C_4	C_5
C_1	**4/5**	0/5	0/5	0/5	0/5
C_2	0/5	**4/5**	0/5	0/5	0/5
C_3	0/5	0/5	**5/5**	0/5	0/5
C_4	1/5	0/5	0/5	**4/5**	0/5
C_5	0/5	0/5	0/5	0/5	**5/5**

Table 3.4: Dataset 2 confusion matrix for Z-PREDICT using RIC.

3.6.2 Results of BPFA with Beta PDF Fitting Features

Two datasets were analyzed with ZB-PREDICT. The results of the BPFA may be seen in Figures 3.7 and 3.8 for Dataset 1 and 2 respectively. For Dataset 1, $K = 50$ was used along with 2000 burn-in iterations and 2000 sample iterations. The Resulting RIC and SIC were 83.3% for both. For Dataset 2, $K = 50$ was used along with 2000 burn-in and 2000 sample iterations. This resulted in 76% RIC and SIC.

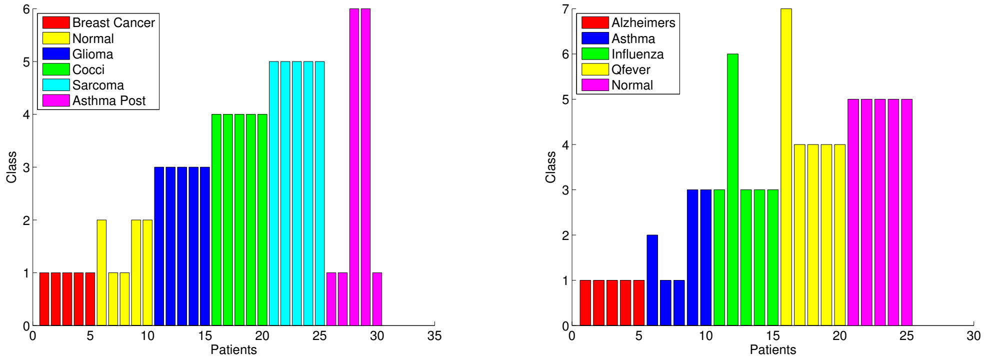


(a) $\hat{\mathbf{\Phi}}$ for Dataset 1

(b) $\hat{\mathbf{Z}}$ matrix transposed, in LOF null features removed for Dataset 1.

(c) $\hat{\pi_k}$ values in descending order

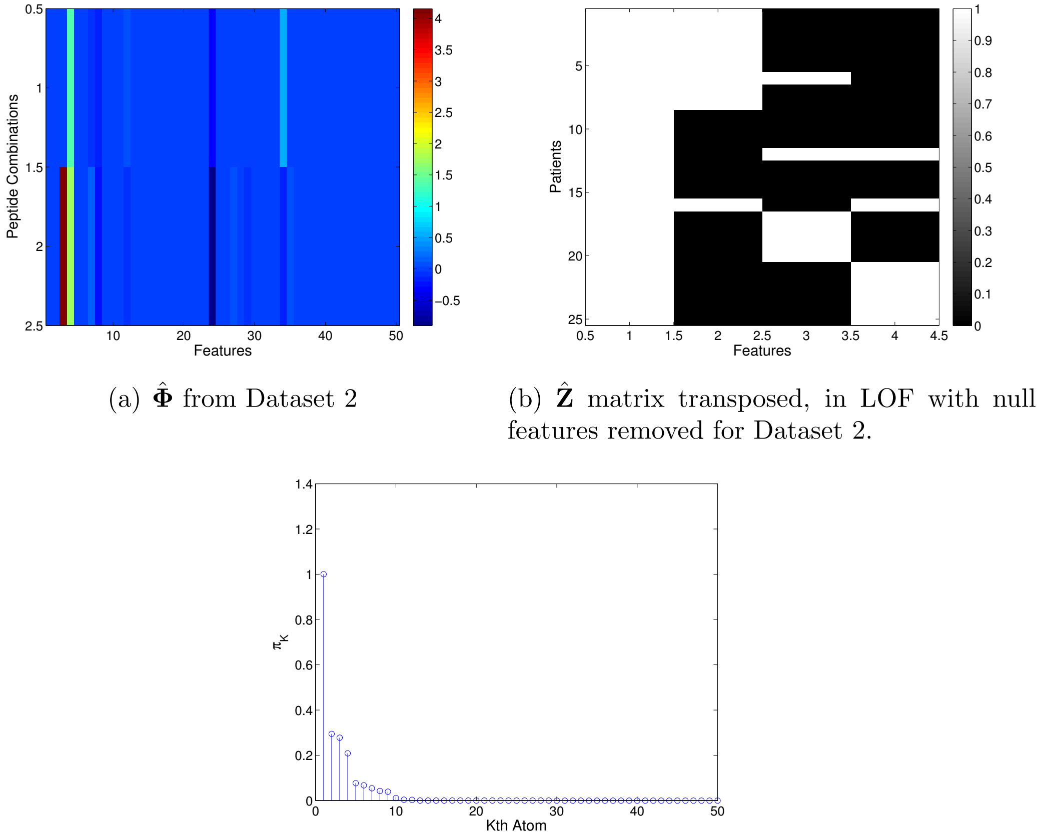
Figure 3.7: ZB-PREDICT BPFA results from Dataset 1



(a) Dataset 1 results.

(b) Dataset 2 results.

Figure 3.9: ZB-PREDICT clustering results.



(a) $\hat{\mathbf{\Phi}}$ from Dataset 2

(b) $\hat{\mathbf{Z}}$ matrix transposed, in LOF with null features removed for Dataset 2.

(c) $\hat{\pi_k}$ values in descending order

Figure 3.8: ZB-PREDICT BPFA results from Dataset 2

Class	C_1	C_2	C_3	C_4	C_5	C_6
C_1	**5/5**	0/5	0/5	0/5	0/5	0/5
C_2	2/5	**3/5**	0/5	0/5	0/5	0/5
C_3	0/5	0/5	**5/5**	0/5	0/5	0/5
C_4	0/5	0/5	0/5	**5/5**	0/5	0/5
C_5	0/5	0/5	0/5	0/5	**5/5**	0/5
C_6	3/5	0/5	0/5	0/5	0/5	**2/5**

Table 3.5: Dataset 1 confusion matrix for ZB-PREDICT using SIC.

Class	C_1	C_2	C_3	C_4	C_5	C_6
C_1	**5/5**	0/5	0/5	0/5	0/5	0/5
C_2	2/5	**3/5**	0/5	0/5	0/5	0/5
C_3	0/5	0/5	**5/5**	0/5	0/5	0/5
C_4	0/5	0/5	0/5	**5/5**	0/5	0/5
C_5	0/5	0/5	0/5	0/5	**5/5**	0/5
C_6	3/5	0/5	0/5	0/5	0/5	**2/5**

Table 3.6: Dataset 1 confusion matrix for ZB-PREDICT using RIC.

Class	C_1	C_2	C_3	C_4	C_5
C_1	**5/5**	0/5	0/5	0/5	0/5
C_2	2/5	**1/5**	2/5	0/5	0/5
C_3	0/5	0/5	**4/5**	0/5	0/5
C_4	0/5	0/5	0/5	**4/5**	0/5
C_5	0/5	0/5	0/5	0/5	**5/5**

Table 3.7: Dataset 2 confusion matrix for ZB-PREDICT using SIC.

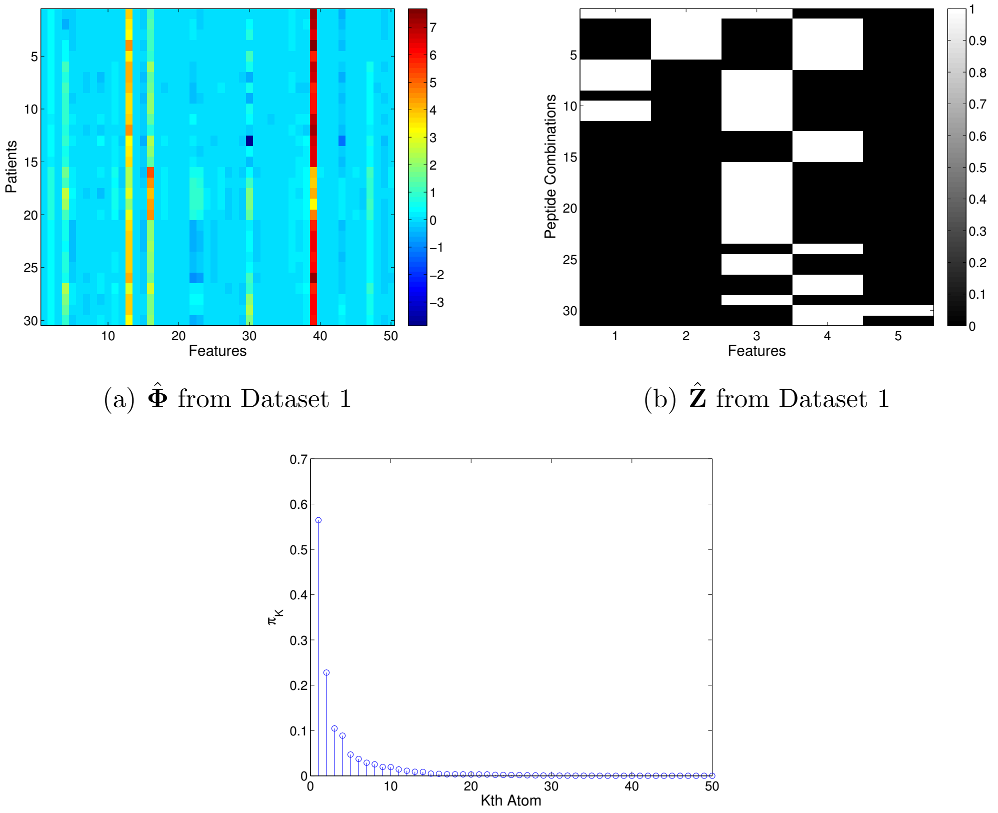
Class	C_1	C_2	C_3	C_4	C_5
C_1	**5/5**	0/5	0/5	0/5	0/5
C_2	2/5	**1/5**	2/5	0/5	0/5
C_3	0/5	0/5	**4/5**	0/5	0/5
C_4	0/5	0/5	0/5	**4/5**	0/5
C_5	0/5	0/5	0/5	0/5	**5/5**

Table 3.8: Dataset 2 confusion matrix for ZB-PREDICT using RIC.

3.6.3 Results of BPFA with Φ Matrix and PCA

As previously stated, data matrix $\mathbf{X}$ is of size DxN. When patients are used as the dimensionality of this matrix, D, this means that the input data matrix must be arranged such that the rows of the matrix contain the N peptide PCA components desired. The simplistic quantization scheme as described in section 3.5.3 was utilized for classification. For these results, RIC is defined as having the exact same features at the exact same quantization levels with the except of a single feature that is within a single quantization level of the rest of the group.

For Dataset 1 the resulting $\hat{\mathbf{\Phi}}$ and $\hat{\mathbf{Z}}$ matrices as well as the estimated $\hat{\pi_k}$ values are reordered and presented in Figure 3.10. The $\hat{\mathbf{Z}}$ matrix is presented in left ordered form (LOF) [92] and are transposed with features containing no entries removed in order to easily show the feature relationships. In this case, it is the $\hat{\mathbf{\Phi}}$ matrix that illustrates the relationship between the patients and underlying features. It is possible to see that the combinations of underlying features are indicative of the disease states of each of the patients. A simple classification scheme is included in Figure 3.11 for dataset 1 that is produced by simple quantization of the values of each entry in the matrix. Also shown in this figure is the comparison of every patient to every other patient for both a scaled and unscaled image. The number of PCA components used in this analysis was 31 which represents 99.93% of the data and $K = 50$ for this analysis. A total of 2500 burn-in iterations and 2500 sample iterations were used for the blocked Gibbs sampler. The confusion matrices for the SIC and RIC cases are given in Tables 3.9 and 3.10. This led to a SIC result of 43.3% and a RIC result of 53.3%. For Dataset 2 the resulting $\hat{\mathbf{\Phi}}$, $\hat{\mathbf{Z}}$, $\hat{\pi_k}$ are presented in Figure 3.12. Once again, $\hat{\mathbf{Z}}$ is presented in the LOF configuration, transposed, and with features having zero entries removed. The simple classification scheme is shown in Figure 3.13 which is produced from simple quantization of the values of each entry in the $\hat{\mathbf{\Phi}}$ matrix. The number of PCA components used in this analysis was 26 which accounts for essentially 100% of the data and $K = 50$ for this analysis. A total of 2000 burn-in iterations and 2000 sample iterations were used. The SIC and RIC confusion matrices are given in Table 4.3 and 4.3 respectively. SIC was 60% while RIC was 68%.

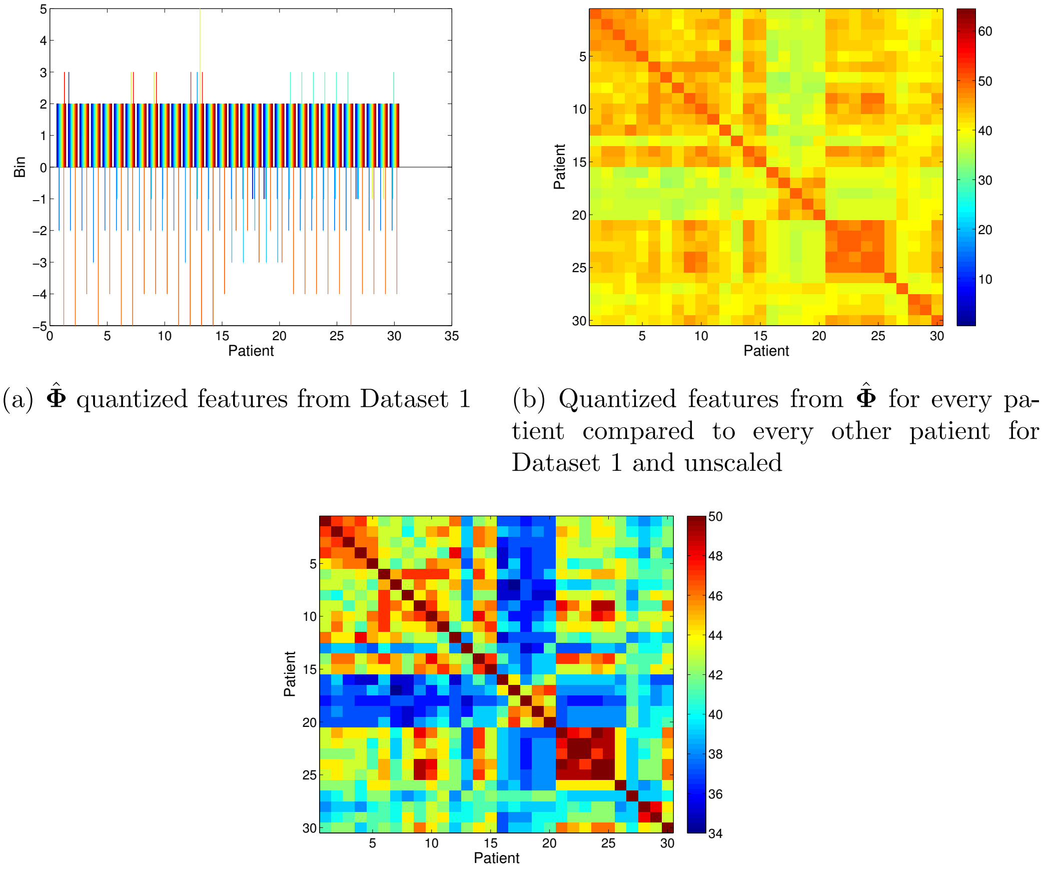


(a) $\hat{\mathbf{\Phi}}$ from Dataset 1

(b) $\hat{\mathbf{Z}}$ from Dataset 1

(c) $\hat{\pi}_K$ atoms in descending order from Dataset 1

Figure 3.10: BPFA output for Dataset 1

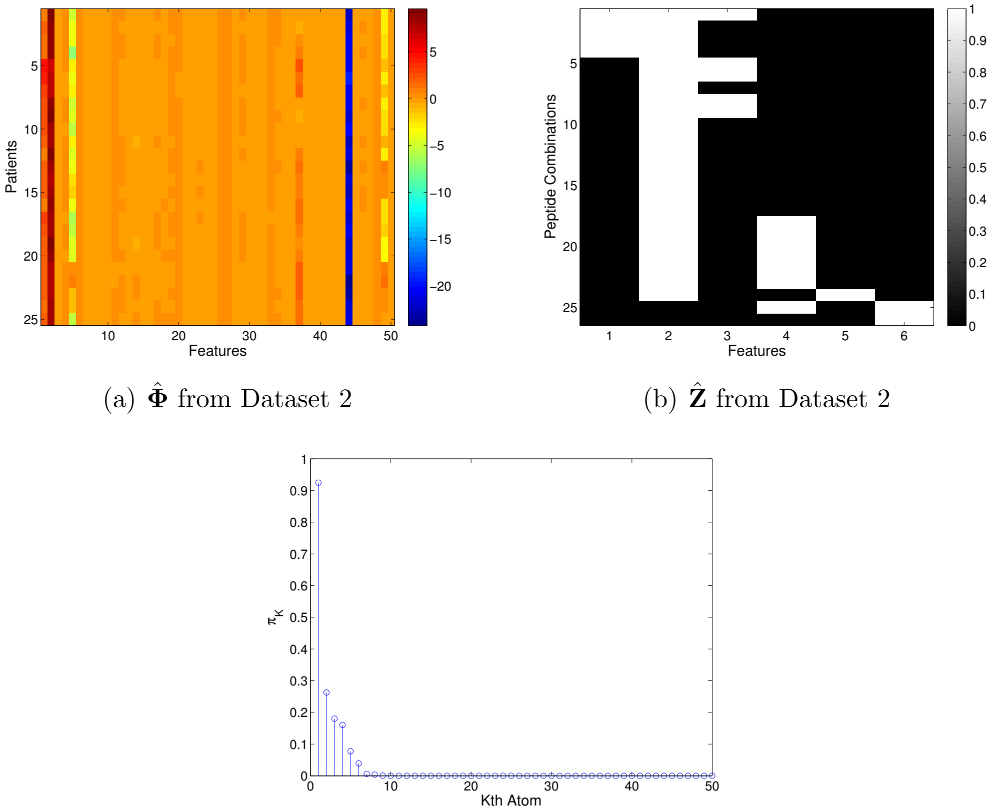


(a) $\hat{\boldsymbol{\Phi}}$ quantized features from Dataset 1

(b) Quantized features from $\hat{\boldsymbol{\Phi}}$ for every patient compared to every other patient for Dataset 1 and unscaled

(c) Quantized features from $\hat{\boldsymbol{\Phi}}$ for every patient compared to every other patient for Dataset 1 and scaled

Figure 3.11: Dataset 1 Φ-PREDICT results

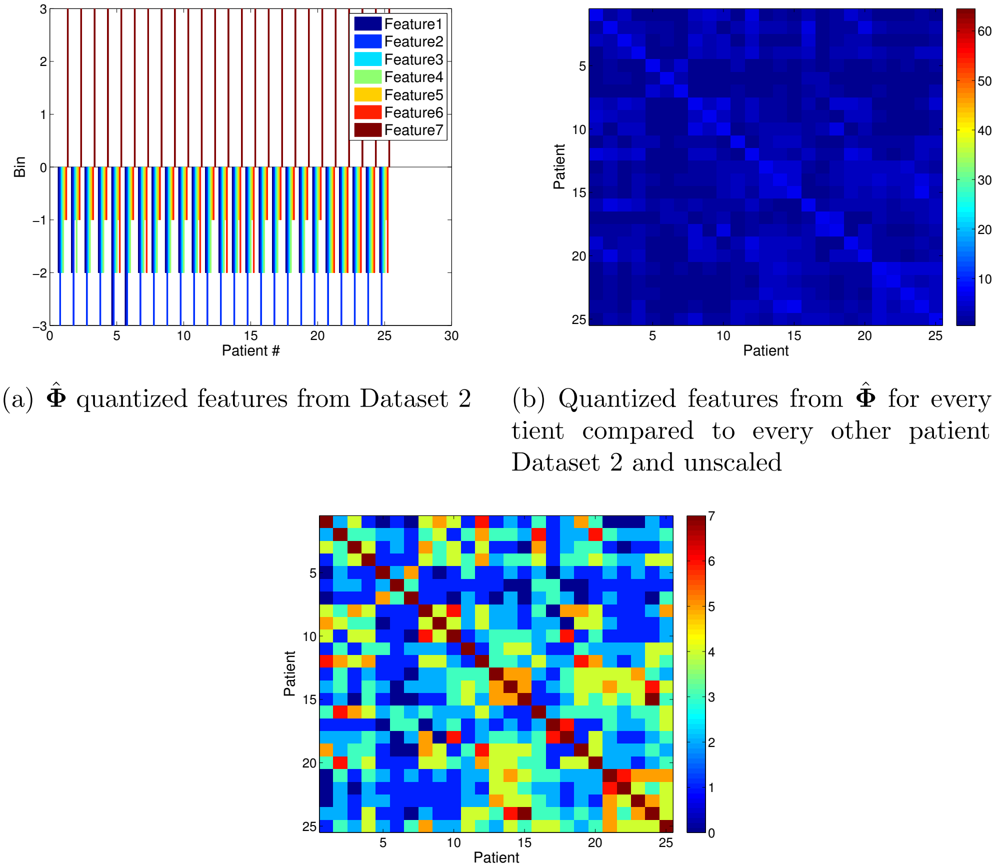


(a) $\hat{\mathbf{\Phi}}$ from Dataset 2

(b) $\hat{\mathbf{Z}}$ from Dataset 2

(c) Descending order $\hat{\pi}_k$ from Dataset 2

Figure 3.12: BPFA output for Dataset 2



(a) $\hat{\mathbf{\Phi}}$ quantized features from Dataset 2

(b) Quantized features from $\hat{\mathbf{\Phi}}$ for every patient compared to every other patient for Dataset 2 and unscaled

(c) Quantized features from $\hat{\mathbf{\Phi}}$ for every patient compared to every other patient for Dataset 2 and scaled

Figure 3.13: Φ-PREDICT results for Dataset 2

Class	C_1	C_2	C_3	C_4	C_5	C_6
C_1	**2/5**	0/5	0/5	0/5	0/5	0/5
C_2	0/5	**1/5**	0/5	0/5	0/5	0/5
C_3	0/5	0/5	**1/5**	0/5	0/5	0/5
C_4	0/5	0/5	0/5	**2/5**	0/5	0/5
C_5	0/5	0/5	0/5	0/5	**5/5**	0/5
C_6	0/5	0/5	0/5	0/5	0/5	**2/5**

Table 3.9: Dataset 1 confusion matrix for Φ-PREDICT results using SIC.

Class	C_1	C_2	C_3	C_4	C_5	C_6
C_1	**2/5**	0/5	0/5	0/5	0/5	0/5
C_2	0/5	**3/5**	0/5	0/5	0/5	0/5
C_3	0/5	0/5	**2/5**	0/5	0/5	0/5
C_4	0/5	0/5	0/5	**2/5**	0/5	0/5
C_5	0/5	0/5	0/5	0/5	**5/5**	0/5
C_6	0/5	0/5	0/5	0/5	0/5	**2/5**

Table 3.10: Dataset 1 confusion matrix for Φ-PREDICT results using RIC.

Class	C_1	C_2	C_3	C_4	C_5
C_1	**1/5**	0/5	0/5	3/5	0/5
C_2	0/5	**1/5**	1/5	3/5	0/5
C_3	0/5	0/5	**4/5**	1/5	0/5
C_4	0/5	0/5	0/5	**5/5**	0/5
C_5	0/5	0/5	1/5	0/5	**4/5**

Table 3.11: Dataset 2 confusion matrix for Φ-PREDICT results using SIC.

Class	C_1	C_2	C_3	C_4	C_5
C_1	**1/5**	0/5	0/5	3/5	0/5
C_2	0/5	**1/5**	1/5	3/5	0/5
C_3	0/5	0/5	**5/5**	0/5	0/5
C_4	0/5	0/5	0/5	**5/5**	0/5
C_5	0/5	0/5	0/5	0/5	**5/5**

Table 3.12: Dataset 2 confusion matrix for Φ-PREDICT results using RIC.

3.7 Challenges with BPFA Clustering

Several challenges to these classification schemes exist, the first of which is the increased computational overhead as the number of patients increases; in other words, a larger **C** matrix is generated. Additionally, as K increases, the **C** values may increase, leading to an artificial inflation of the values simply due to more shared features. Thus, it will not be possible to compare **C** values from different dataset runs when new BPFA results are generated, though this may be solved by normalizing the resulting feature vectors. Additionally, the quantization scheme for Φ-PREDICT is user specified and its performance is dependent on the appropriate level being employed. If there are too many quantization levels used, then it is less likely that patients with similar diseases will have the exact same features at the exact same quantization levels due to too fine resolution. If there are too few quantization levels, there is the possibility that patients with different diseases will have similar feature representations at identical quantization levels due to too poor resolution.

Furthermore, extra care is needed when determining what will be considered correct classification using this method. Definitions for SIC are fairly straightforward, but when considering RIC, the user must select appropriate amounts of variation that will be considered allowable in the final classification analysis. This is often tied to the classification parameters selected such as number of quantization levels or number of features considered. As such, this creates a highly variable picture as to what can be considered correct classification, and may need to be considered on a case by case basis, depending on how the output will be used. This can also lead to different interpretations for SIC and RIC criteria as well. For example, it might be perfectly acceptable in a research or discovery setting to allow for some variation in the feature quantization levels, where as in the clinical setting, one may want to go with the SIC definition as clinical action will be taken based on the results.

For this reason, matrices comparing each output to that of every other output are provided for easier results viewing. It is possible that further analysis on these comparison matrices (such as thresholding for what will be considered correct classification) would yield additional insights into the classification results. Additionally, all of these methods are non-adaptive and do not easily allow additional patient data to be analyzed; this method was not further explored within the context of this work.

While the BPFA is very useful in highlighting underlying feature relationships, it does not inherently perform classification for immunosignaturing data based on the less restrictive definition that multiple features may describe a disease state. For example, once having found the $\hat{\mathbf{\Phi}}$ and $\hat{\mathbf{Z}}$ matrices in each case, it may not be intuitively obvious which patients should be grouped together. This is especially difficult as the number of features increases. As such, further classification algorithms may still be needed to illustrate the patient groupings. Additionally, the number of estimated parameters is much greater than in the DP-GMM based methods previously discussed, which leads to increased computational complexity and execution time. While improvements in these areas are not the focuses of this work, it is possible to further refine the algorithms, such as by the implementation of parallel architectures, to improve the BPFA execution.

With regards to the BPFA model parameters themselves, the prior distribution assumptions were also selected in this case to provide some nice conjugate prior simplifications. Other distributions may be selected that could be more appropriate to the data types and improve the subsequent classification. Additionally, the innovation parameters a and b (and the equation related K) play a role in how latent features are grouped together. Therefore, it is possible to select these parameters such that features are not usually grouped together or such that they are very often grouped together. The user input is somewhat critical for these parameters, and they should be carefully selected based on how conservative one needs to be in terms of missclassification.

Chapter 4

INTEGRATED DP-GMM AND BPFA CLUSTERING

4.1 Integration of DP-GMM Clustering with BPFA Latent Feature Analysis

While beta process factor analysis (BPFA) allows for the determination of underlying features that describe the patient, disease, antibody, and peptide relationships measured by immunosignaturing, it requires a clustering scheme in order to group the latent features. This is because clustering is based on combinations of features rather than the more restrictive assumption of one feature corresponding to one cluster only. Although the heuristic clustering schemes discussed in Chapter 3 using the $\hat{\mathbf{\Phi}}$ and modified $\hat{\mathbf{Z}}$ matrices as estimations of Equation (3.8) were successfully used, their expansibility and utility is somewhat limited. This is because the heuristic clustering algorithms become increasingly difficult to use as the number of features, patients, or patient data increases. As such, an adaptive method like the DP-GMM may be used to perform the clustering.

A combined approach using the Dirichlet Process (DP) and BPFA is used to learn a dictionary for image construction in [98], but does not focus on identifying latent features. In [46], an infinite overlapping mixture model was used by assigning data to multiple clusters following the Indian buffet process (IBP), assuming underlying features as clusters, but this is too restrictive for immunosignatures. In [99], a combined beta process DP model was considered for a compressive sensing framework, but it doesn't consider both visible and latent processes. A further use of the DP with the IBP is described in [45], which assumes that there are multiple clustering interpretations for the resulting features rather than a single grouping.

4.1.1 DP-GMM and BPFA Clustering Using PCA Features

The same DPGMM algorithm presented in Algorithm 1 may be used to adaptively classify both the $\hat{\mathbf{\Phi}}$ or the $\hat{\mathbf{Z}}$ matrices in order to show patient groups that indicate similar disease states. The BIO-PREDICT, or **B**PFA **I**ncluded pr**O**cessing with **P**CA **RE**solution and **D**PGMM for **I**mmunosignature **C**lassification **T**esting, combines PCA dimensionality reduction with the BPFA for latent feature identification, and then feeds the result of the $\hat{\mathbf{Z}}$ matrix into the DPGMM for adaptive clustering. In terms of immunosignaturing, this method provides a flexible, adaptive method for on-the-fly clustering that is applicable in the situation where additional data for new patients is received, or when additional immunosignaturing data for existing patients is received. This method does not require any modification of the existing BPFA or DPGMM algorithms, making it a good fit and logical next step for the immunosignaturing clustering problem. The combination of these two methods allows for complex feature relationships that are algorithmically determined via Bayesian nonparametrics rather than described by only the observable data characteristics available to the researcher. Additionally, since the clustering is adaptive, it is able to update model parameters and adjust to new incoming data.

4.1.2 DP-GMM and BPFA Clustering Using Beta PDF Fitting Features

Just as PCA was used for dimensionality reduction in BIO-PREDICT, the beta probability density function fitting (BPF) can be used to obtain the BIOB-PREDICT, or **B**PFA **I**ncluding pr**O**cessing using **B**eta **P**DF **RE**solution with **D**PGMM for **I**mmunosignature **C**lassification **T**esting algorithm. This approach uses beta PDF fitting for feature reduction, followed by BPFA for latent feature identification, and a reduced and transposed $\hat{\mathbf{Z}}$ matrix is then fed into the DP-GMM for adaptive clustering. This method provides an alternative to PCA in the BIO-PREDICT method, thereby limiting the feature space to two dimensions for all patient samples.

4.2 Clustering Results for Integrated Clustering Approaches

4.2.1 Results of DP-GMM with BPFA and PCA

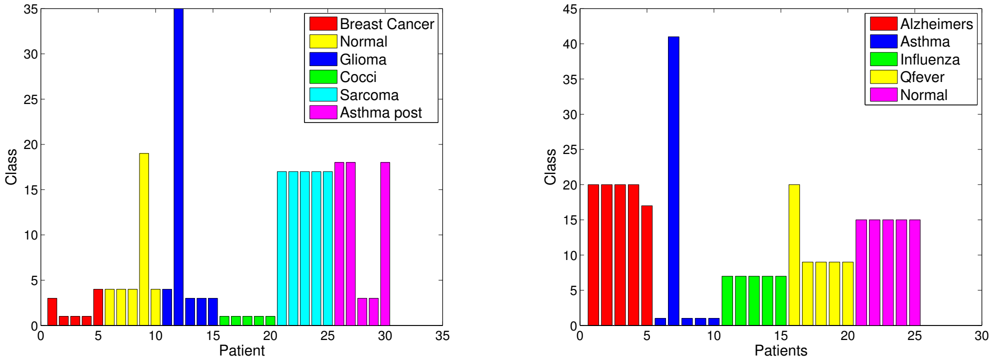
Several datasets were analyzed using the BIO-PREDICT method. For Dataset 1, the Z-PREDICT BPFA results given in Figure 3.4 were used as the input for BIO-PREDICT (as opposed to running the same data from the PCA stage). For the remaining DPGMM stage, $\alpha = 35$ was chosen, the number of burn-in iterations was 2000, and the number of sample iterations was 3000. For the Dataset 2, the BPFA results of Z-PREDICT in Figure 3.5 were used. The remaining DPGMM step used $\alpha = 45$, and 2000 burn-in iterations and 3000 sample iterations were used. The results of the clustering are seen in Figures 4.1(a) and 4.1(b), and the corresponding confusion matrices are seen in Tables 4.1, 4.2, 4.3, and 4.4. The clustering result for this was found to be 76.7% for both SIC and RIC for Dataset 1 and 88% for both SIC and RIC for Dataset 2.

Class	C_1	C_2	C_3	C_4	C_5	C_6
C_1	**0/5**	1/5	1/5	3/5	0/5	0/5
C_2	0/5	**4/5**	0/5	0/5	0/5	0/5
C_3	0/5	1/5	**3/5**	0/5	0/5	0/5
C_4	1/5	0/5	0/5	**5/5**	0/5	0/5
C_5	0/5	0/5	0/5	0/5	**5/5**	0/5
C_6	0/5	0/5	2/5	0/5	0/5	**3/5**

Table 4.1: Dataset 1 confusion matrix for BIO-PREDICT using SIC.

Class	C_1	C_2	C_3	C_4	C_5	C_6
C_1	**0/5**	1/5	1/5	3/5	0/5	0/5
C_2	0/5	**4/5**	0/5	0/5	0/5	0/5
C_3	0/5	1/5	**3/5**	0/5	0/5	0/5
C_4	1/5	0/5	0/5	**5/5**	0/5	0/5
C_5	0/5	0/5	0/5	0/5	**5/5**	0/5
C_6	0/5	0/5	2/5	0/5	0/5	**3/5**

Table 4.2: Dataset 1 confusion matrix for BIO-PREDICT using RIC.



(a) BIOPREDICT results from Dataset 1

(b) BIOPREDICT results from Dataset 2

Figure 4.1: BIO-PREDICT results

Class	C_1	C_2	C_3	C_4	C_5
C_1	**4/5**	0/5	0/5	0/5	0/5
C_2	0/5	**4/5**	0/5	0/5	0/5
C_3	0/5	0/5	**5/5**	0/5	0/5
C_4	1/5	0/5	0/5	**4/5**	0/5
C_5	0/5	0/5	0/5	0/5	**5/5**

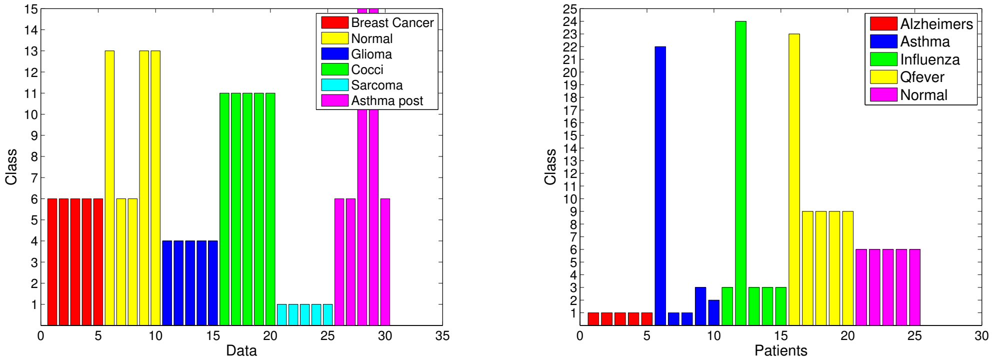
Table 4.3: Dataset 2 confusion matrix for BIO-PREDICT using SIC.

Class	C_1	C_2	C_3	C_4	C_5
C_1	**4/5**	0/5	0/5	0/5	0/5
C_2	0/5	**4/5**	0/5	0/5	0/5
C_3	0/5	0/5	**5/5**	0/5	0/5
C_4	1/5	0/5	0/5	**4/5**	0/5
C_5	0/5	0/5	0/5	0/5	**5/5**

Table 4.4: Dataset 2 confusion matrix for BIO-PREDICT using RIC.

4.2.2 Results of DP-GMM with BPFA and Beta PDF Fitting

The BIOB-PREDICT algorithm was performed using the BPFA outputs given in Figures 3.7 and 3.8. For the remaining DPGMM step for Dataset 1, $\alpha = 35$, 2000 burn-in iterations, and 3000 sample iterations were used. This results in a SIC and RIC clustering rate of 83.3% in both cases. For Dataset 2, the remaining DPGMM steps used $\alpha = 35$, 2000 burn-in iterations, and 2000 sample iterations. For this data set, the SIC and RIC clustering results were both 76%.



(a) BIOBPREDICT results from Dataset 1. (b) BIOBPREDICT results from Dataset 2.

Figure 4.2: BIOB-PREDICT clustering results.

Class	C_1	C_2	C_3	C_4	C_5	C_6
C_1	**5/5**	0/5	0/5	0/5	0/5	0/5
C_2	2/5	**3/5**	0/5	0/5	0/5	0/5
C_3	0/5	0/5	**5/5**	0/5	0/5	0/5
C_4	0/5	0/5	0/5	**5/5**	0/5	0/5
C_5	0/5	0/5	0/5	0/5	**5/5**	0/5
C_6	3/5	0/5	0/5	0/5	0/5	**2/5**

Table 4.5: Dataset 1 confusion matrix for BIOB-PREDICT using SIC.

Class	C_1	C_2	C_3	C_4	C_5	C_6
C_1	**5/5**	0/5	0/5	0/5	0/5	0/5
C_2	2/5	**3/5**	0/5	0/5	0/5	0/5
C_3	0/5	0/5	**5/5**	0/5	0/5	0/5
C_4	0/5	0/5	0/5	**5/5**	0/5	0/5
C_5	0/5	0/5	0/5	0/5	**5/5**	0/5
C_6	3/5	0/5	0/5	0/5	0/5	**2/5**

Table 4.6: Dataset 1 confusion matrix BIOB-PREDICT using RIC.

Class	C_1	C_2	C_3	C_4	C_5
C_1	**5/5**	0/5	0/5	0/5	0/5
C_2	2/5	**1/5**	1/5	0/5	0/5
C_3	0/5	0/5	**4/5**	0/5	0/5
C_4	0/5	0/5	0/5	**4/5**	0/5
C_5	0/5	0/5	0/5	0/5	**5/5**

Table 4.7: Dataset 2 confusion matrix BIOB-PREDICT using SIC.

Class	C_1	C_2	C_3	C_4	C_5
C_1	**5/5**	0/5	0/5	0/5	0/5
C_2	2/5	**1/5**	1/5	0/5	0/5
C_3	0/5	0/5	**4/5**	0/5	0/5
C_4	0/5	0/5	0/5	**4/5**	0/5
C_5	0/5	0/5	0/5	0/5	**5/5**

Table 4.8: Dataset 2 confusion matrix for BIOB-PREDICT using RIC.

4.3 BIO-PREDICT and BIOB-PREDICT Model Challenges

For BIO-PREDICT, the PCA steps require correct eigenvalue truncation, as well as the assumption that the data can be maximized along orthogonal basis. The modified covariance matrix also requires an understanding of the underlying data in order to select an appropriate shrinkage factor. These challenges require additional user analysis and input. For the BIOB-PREDICT algorithm, the BPF step results in a large amount of data loss when reducing to two parameters, and assumes that the feature space is sufficient to capture subtle antibody differences. For both BIO-PREDICT and BIOB-PREDICT, while using the DP-GMM is an improvement over the proposed simple clustering schemes in that it is adaptive, it still has its own challenges (see section 2.6). The issues with innovation parameter selection, underlying distribution selection (Gaussian), and single cluster membership are tradeoffs. Additionally, the BPFA steps require the selection of a distribution and feature parameters, and while a version is presented here, other distributions may be more suited to the data. Thus, both methods can prove to be computationally intensive when evaluated using the blocked Gibbs sampler method in a Markov chain Monte Carlo framework.

Chapter 5

MULTIPLE STATE CLUSTERING USING BPFA

5.1 Motivation for Multiple Disease State Clustering

It has been shown that it is possible to model and cluster immunosignaturing data using Bayesian nonparametric techniques such as Dirichlet process Gaussian mixture modeling (DP-GMM) and the beta process factor analysis (BPFA) when these are incorporated into a broader algorithm flow. The previously presented clustering algorithms assume only patients with a single state, or single underlying disease. However, there may be instances when clustering or classification into multiple underlying states is necessary. For example, a patient may suffer from more than one disease or multiple strains of the same disease [40].

There is also a desire not only to determine patient pathology groupings, but also to determine pathology combinations in patients (i.e., the presence of multiple diseases). Other possible examples include the desire to identify a single disease as well as the disease stage, the need to illustrate single disease relationships where multiple relationships are possible with other diseases, and the desire to know both diseases and symptoms. Additionally, there may be relationships between single diseases or disease manifestations (such as expanded relaxed immunosignaturing classification, RIC, criteria) that are not explored when a single group is assigned. As such, additional algorithms need to be developed to allow multiple cluster membership for a single patient sample. Furthermore, while clustering was successful in previous results, there was no disease identification determined; results are restricted to only the basic groupings amongst patients.

There is a desire to extend the single state techniques to accommodate for multiple underlying states and provide flexibility for further analysis. We propose to modify the heuristic Z matrix based clustering algorithms in Chapter 3 for determining multi-state relationships amongst patients by introducing known information for desired states [48]. This may also be thought of as training data required for state identification but is not used for clustering. In the case of disease data, this would mean immunosignatures that are associated with a known disease state.

5.2 Algorithms for Multiple State Clustering Based on BPFA

5.2.1 PCA Features with BPFA for Multiple State Clustering

Previously, it was demonstrated that it is possible to separate patient groups using the Z-PREDICT algorithm and further using the BIO-PREDICT algorithm to facilitate easier clustering. The inputs in both of these cases were log PCA data based on individual patient immunosignatures consisting of the median intensities of peptide sequences on the microarray. The combinations of latent features were used to define a single disease state, where more than one feature may be indicative of a particular patient group. However, these algorithms will only indicate underlying feature combinations; they do not directly separate out which feature combination pertains to which disease unless a training set of data is available. This training data, or disease key data, is necessary for diagnosis in order to establish the baseline responses for known states.

In order to expand Z-PREDICT to be useful for n underlying states, some equation modifications need to be made. As such, the Z-PREDICTn, or **Z**-matrix from **P**CA **RE**solution and **D**iscovery for **I**mmunosignature **C**lassification **T**esting up to **n**-states, algorithm is developed. In this method, a master key for single states (i.e., training data for the single state) is used to compare against the individual data entries with multiple underlying states. The algorithm is built upon the basis that the latent features for multi-state patients are some mathematical operation between the feature vectors for the individual states. For example, note that Patient 1 (a representative disease key) in Figure 5.1 has a feature vector of $\mathbf{z_1} = [110000]$ and Patient 15 (another disease key) has a feature vector of $\mathbf{z_{15}} = [101100]$. Each of these is a feature vector for a separate disease. Compare this to Patient 9 whose feature vector is $\mathbf{z_9} = [111100]$ and whose patient contains both diseases that are represented by $\mathbf{z_1}$ and $\mathbf{z_{15}}$, which can be seen as a logical operation union between the two vectors, notably the "or" function in this case.

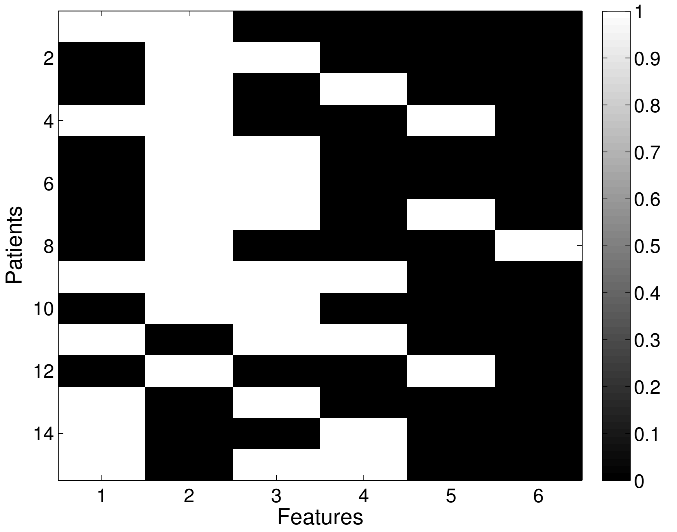


Figure 5.1: Example of dual disease state data $\hat{\mathbf{Z}}$ (LOF and transposed)

To create a method to exploit this, a modification of binary BPFA matrix $\hat{\mathbf{Z}}$ (the estimate of the binary matrix in Equation (3.8)) is necessary. The matrix of single state keys is known as the $\mathbf{M}$ matrix, and consists of a single feature vector for each known state, where the feature vector is meant to represent the weighted values of each feature within a representative population with only that state. The $\hat{\mathbf{Z}}$ matrix then needs to be modified such that incorrect comparisons will be penalized. This is accomplished by replacing all zero entries from the original BPFA output with a negative one. This is known then as the matrix $\mathbf{Y}$. The compete absence of a feature is denoted by -1 and strong presence of a feature is denoted as 1. The master keys for the single disease are then appended to $\mathbf{Y}$, and the new matrix is referred to as $\mathbf{A}$. This matrix is then multiplied by the transpose of itself, and the upper-triangular portion is kept while the lower-triangular portion is set to zero, and only the first R rows are kept. This can be described as a new $\mathbf{Z}$ matrix modification and comparison scheme between the known key data and the unknown patient data with the following mathematical representations:

$$\mathbf{Y} = g.(2\mathbf{Z} - \mathbf{1}) \tag{5.1a}$$

$$\mathbf{Y} = [\mathbf{N}; \mathbf{P}] \tag{5.1b}$$

$$\mathbf{M}(r) = \sum_{1}^{q} p_{q,r}\mathbf{N}(r), r = 1 \ldots R \tag{5.1c}$$

$$\mathbf{A} = [\mathbf{M}; \mathbf{P}] \tag{5.1d}$$

$$\mathbf{C} = \mathrm{UT}[(h\mathbf{A})^T(h\mathbf{A}) + c, R] \tag{5.1e}$$

Where $\mathrm{UT}[., R]$ keeps only the upper triangular portion of the matrices that correspond to the desired R rows, R is the number of master keys (i.e., the number of rows) in $\mathbf{M}$, $\mathbf{P}$ is the portion of $\mathbf{Z}$ that contains only entries with unknown states and no key results, $\mathbf{N}$ is the portion containing only the non-averaged key results, p is the number of results in $\mathbf{Z}$ that correspond to each state, and g and h are scaling factors and c is a constant that helps to account for very small number multiplication if necessary. Additionally, q is the total number of patients in that particular state and their individual weights are $p_{q,r}$. In general, the parameter $p_{q,r}$ can be adjusted to account for varying amounts of competing antibodies, in the case of immunosignaturing. This can also be thought of one possible way to incorporate information surrounding inhibition, such as competitive inhibition [101, 102]. This Z-PREDICTn method may also be thought of as a non-normalized scaled dot product or correlation. An alternative representation which does not average together keys for known states may be obtained by omitting Equations (5.1c) and (5.1d) with $\mathbf{Y} = \mathbf{A}$. This is similar to the procedure for Z-PREDICT, but may lead to a more complex representation of the classification. However, this can help account for instances where entries are repeated in the training data, or known keys. This difference will be discussed further in the data simulations.

For immunosignaturing, the single states are individual diseases and the multi-state case is when a patient immunosignature contains antibody representations indicative of multiple underling diseases. As as example of this process, it is assumed that the ground truth, or single state known keys, of patients 1-5 (breast cancer) and 11-15 (glioma) are known of Figure 5.1. Thus, in order to provide a single vector describing each particular disease state, the entries are averaged together, creating disease "master keys." This may be seen in Figure 5.2, where the first row is the average of all of the breast cancer feature responses, and row 2 is the average of all the glioma feature responses. They are referred to as the ground truth disease master keys for these two disease states.

Based on results given in Figure 5.2, comparisons may now be made between each remaining patient (rows 3-7) and the two ground truth disease keys given in rows 1 and 2 by using Z-PREDICTn. This comparison (i.e. $\mathbf{C}$) is given in Table 5.1, when all scaling factors of Z-PREDICTn are set to 1. Note that K1 and K2 refer to rows 1 and 2 in Figure 5.2 corresponding to breast cancer and glioma ground truth keys respectively. Also note that R3-R7 refer to rows 3-7 of Figure 5.2 corresponding to each of the five patients with two disease states. Note that negative numbers in the

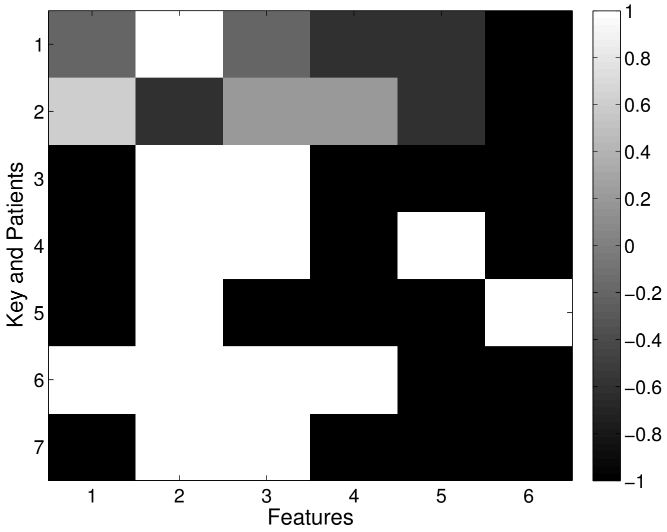


Figure 5.2: Dual disease state data $\hat{\mathbf{Z}}$ (LOF and transposed) and ground truth Key

resulting **C** matrix values show little similarity between the key and patient feature vector of interest and positive numbers show higher similarity. From this matrix it is possible to see that all patients (R3-R7) show positive **C** values with K1, indicating that they most likely share the disease state breast cancer. The relative strengths are given by the **C** values. Similarly **C** values of all but two patients are positive, indicating that they also most likely have glioma. Again, the relative strengths are given by their **C** values. In this example, this would result in correct classification of all 5 patients with K1 and 3 patients with K2.

Key	**K1**	**K2**	**R3**	**R4**	**R5**	**R6**	**R7**
Breast Cancer (K1)	2.80	0.48	3.20	2.00	1.60	1.60	3.20
Glioma (K2)	0.48	2.16	0.40	-0.80	-2.00	2.00	0.40

Table 5.1: **C** matrix results from Figure 5.2

5.2.2 Beta PDF Fitting Features with BPFA for Multiple State Clustering

Similarly to how PCA may be used for dimensionality reduction in Z-PREDICTn, beta PDF fitting (BPF) may be be used as an alternative. This method is called ZB-PREDICTn, or **Z**-matrix from **B**eta **P**DF **RE**solution and **D**iscovery for **I**mmunosignature **C**lassification **T**esting up to **n**-states, and is used when $n \geq 2$. This method uses BPF reduction fed into BPFA for latent feature identification. The $\hat{\mathbf{Z}}$ matrix is then modified according to Equation (5.1). The **C** matrix then describes the relationships. Again, with the immunosignaturing work discussed here, a state is a single disease while multiple states is indicative of a patient having multiple diseases. However, for other work any number of concurrent unknown underlying states does not change the ZB-PREDICTn algorithm execution.

5.3 Multiple Disease State Clustering Results

5.3.1 Multi-Disease Dataset Descriptions

Two sets of disease datasets containing patients with single and dual disease pathologies are also considered. The first group, labeled as multi-disease Dataset 1, or MDD1 consists of 20 sets of key (or ground truth) immunosignatures, five from each of the disease states: *breast cancer, sarcoma, glioma,* and *normal.* Then, ten sets of immunosignatures were used for classification. The first five contained immunosignatures corresponding to breast cancer and sarcoma, while the remaining five contained immunosignatures corresponding to sarcoma and glioma. Immunosignatures were placed in the following order for the analysis: 5 breast cancer, 5 breast cancer + sarcoma, 5 sarcoma, 5 sarcoma + glioma, 5 glioma, and 5 normal. Each dual disease immunosignature was created by taking the average of two different immunosignatures, one from each disease, where median intensities of corresponding sequences were averaged together. The second set of data, known as multi-disease dataset 2 or MDD2, consisted of 59 sets of key immunosignatures, 20 Alzheimer's immunosignatures (the first 20 entries), 20 cocci immunosignatures (entries 41 through 60), 10 myeloma immunosignatures (entries 61-70), and 9 normal immunosignatures (entries 71-79). The multi-disease immunosignatures (entries 21-40) had both Alzheimer's disease and myeloma.

Again, each dual disease immunosignature was created by taking the average of two different immunosignatures, one from each disease, where the median intensities of corresponding sequences were averaged together. Note that, where possible, different immunosignatures were used than what are represented in the training data, but due to small dataset availability for some diseases and the desire to show the algorithm functioning under various conditions, some were repeated in the sarcoma, Alzheimer's, and myeloma groups. However, this does not greatly impact the results for the algorithm since each repetition represents only a small portion of the training data used in each comparison. Analysis and discussions on this averaging with relationship to the biological model are provided in Section 7.1.

5.3.2 Type 1 and 2 Clustering Errors for Multi-disease Data

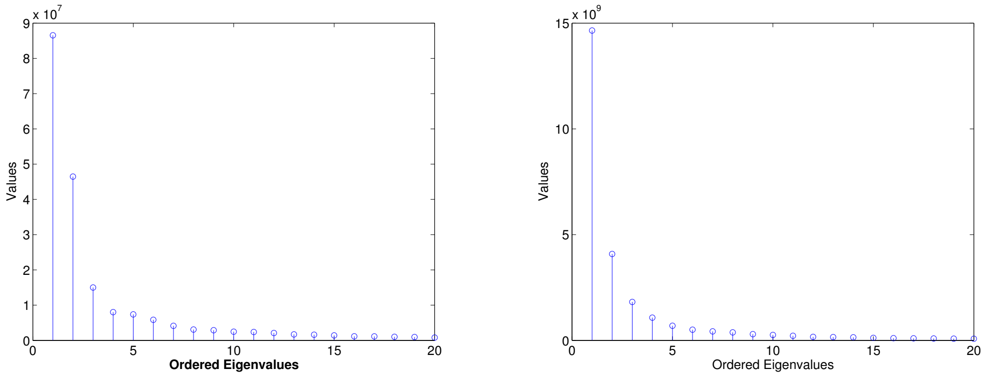
For single disease state clustering results are described using relaxed (RIC) and strict (SIC) definitions. However, when analyzing classification results for multi-disease datasets, it is possible to have multiple types of misclassifications where some diseases may be classified correctly and others may not. Type I and Type II error definitions may be used for this patient data. Statistically speaking, the Type I error generally describes the outcome of incorrectly rejecting a null hypothesis (false positive) [100]. In the case of immunosignaturing, a false positive occurs when a patient has a disease but the classification fails to indicate the presence of this disease.

The Type II error generally refers to the failure to reject a null hypothesis (false negative) [100]. For immunosignaturing classification, this means that a patient is classified as not having a particular disease when in fact they do have that disease pathology.

While these same definitions can be applied to the single disease states as well, it is more straightforward to report the cases of correct classification rather than delineate between the different error cases. However, for dual disease state data, it becomes necessary as some algorithm parameters may work better for certain disease states. As such, multi disease state data will be reported by indicating the true positive, true negative, Type I error, and Type II error for each disease state. The true positive and true negative rates will be combined and referred to as correct classification and the Type I and Type II error rates will be combined and referred to as incorrect classification [100]. When using adaptive clustering, the final category, no result, indicates that the clustering fell outside of the main results and disease presence could not be automatically determined based on the classification results alone without additional analysis by the user. To provide a conservative estimation of the performance, the no result category is considered a misclassification.

5.3.3 Multiple Disease Clustering Results using PCA and BPFA

The Z-PREDICTn algorithm was evaluted with two datasets, MDD1 and MDD2. The eigenvalue plots for MDD1 and MDD2 may be seen in Figure 5.3. The first 11 principal components were used for MDD1, representing 93.2% of the data, while the first 15 principal components were used for MDD2, representing 94.7% of the data. The post BPFA results for the PCA of MDD1 and MDD2 may be seen in Figure 5.4 and Figure 5.5. For both datasets, $K = 50$, and 2000 burn-in and sample iterations each were used.



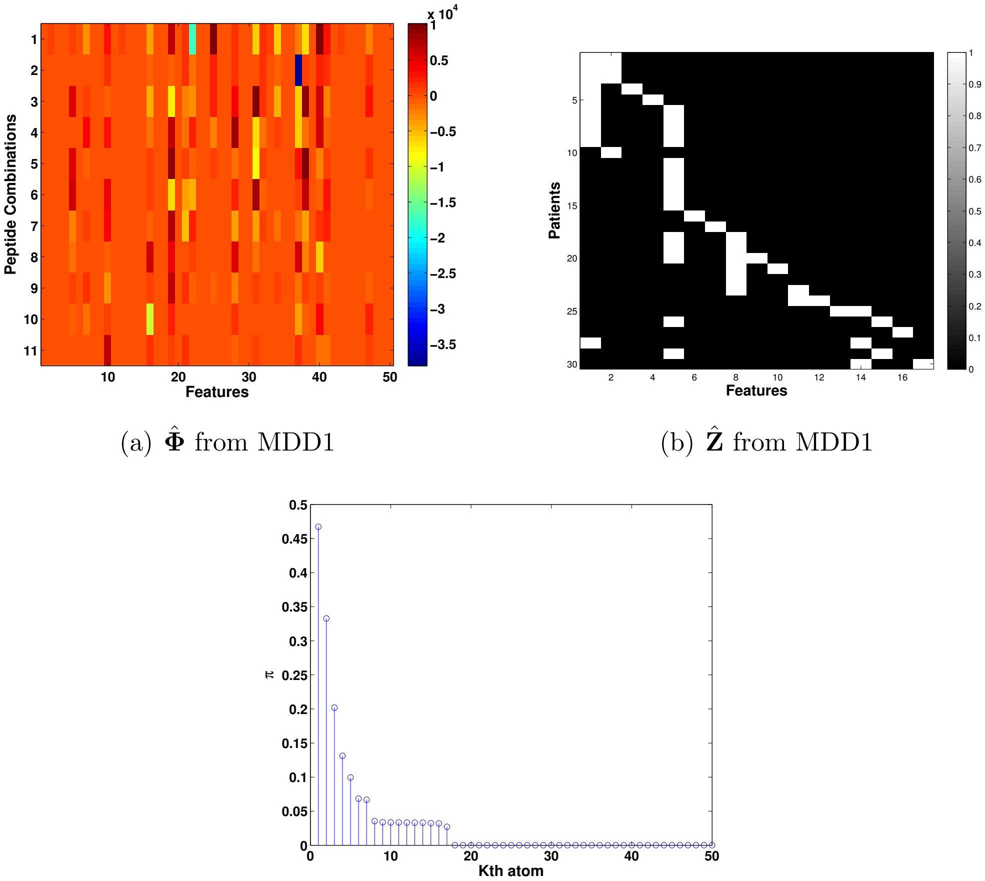
(a) First 20 eigenvalues of MDD1 PCA.

(b) First 20 eigenvalues of MDD2 PCA.

Figure 5.3: The first 20 significant eigenvalues from each multi-disease dataset.

For MDD1, the **C** values are given in Table 5.2 and also plotted in Figure 5.6 for readability. Figure 5.8 and Table 5.3 correspond to MDD2. It should be noted that row 5 (Key 5) in both datasets is simply blank and is inserted for readability only. As one can see from Figure 5.6 and 5.8, the higher the value, the more likely the patient is to have a particular disease. The patient entries for **C** are presented as a stem plot for readability with the x-axis representing each patient and each color coded stem in the plot representing the comparison with the known disease keys. For each dataset, $g = 10$ and $h = (1/100)^{1/2}$.

However, without knowing how many states, i.e. diseases, are present, it becomes difficult to provide a meaningful threshold of where disease significance begins and therefore determine which diseases are present. Thus, in order to determine how many true positive, true negative, type I error, and type II error classification results there are, it is necessary to know the value of n where $n > 1$ and is the number of states (i.e. diseases) of interest. However this is not possible without some supplemental patient knowledge. It is assumed that the value of n is known ($n = 2$ in both cases here). While this determination may be done via several means, it is outside this chapter's scope; an alternative approach will be discussed in Chapter 6. Based



(a) $\hat{\mathbf{\Phi}}$ from MDD1

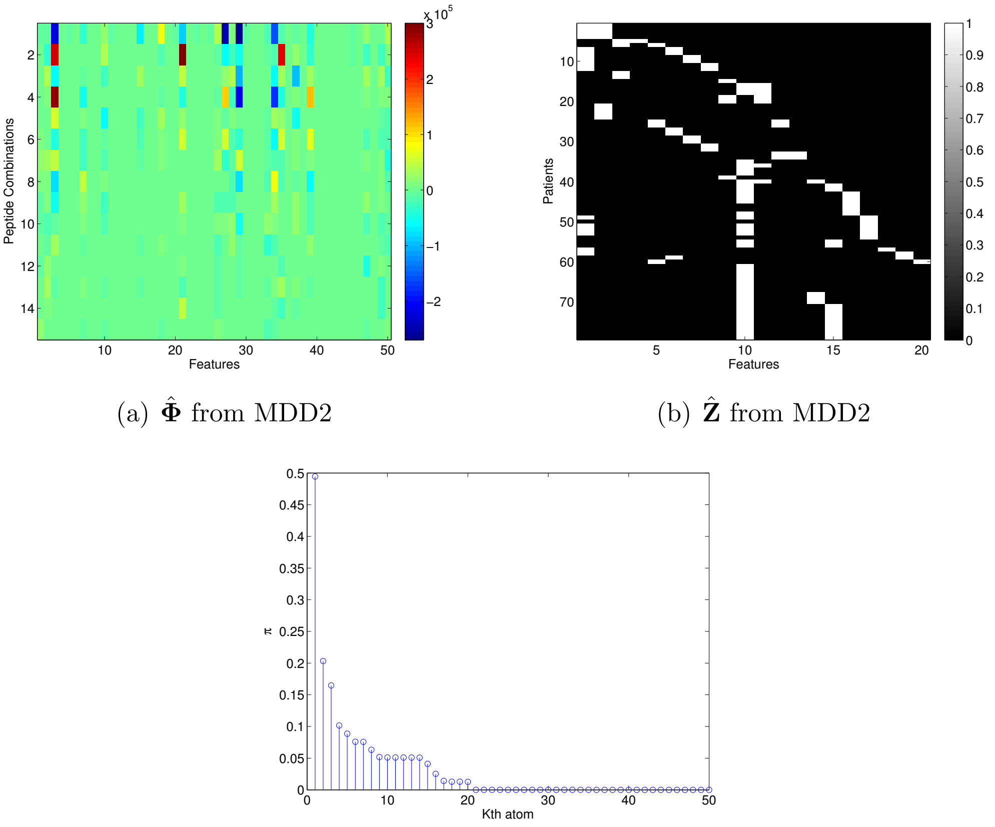
(b) $\hat{\mathbf{Z}}$ from MDD1

(c) $\hat{\pi}_k$ atoms in descending order from MDD1

Figure 5.4: BPFA results for MDD1 in Z-PREDICTn

on this assumption, the classification results in each case are summarized in Tables 5.10 and 5.11. For MDD1 this led to a true combined classification rate of 95% and a combined error classification rate of 5%. For MDD2, this led to a combined true classification rate of 87.5% and a combined error classification rate of 12.5%.

Also given are the un-combined results where Equations (5.1c) and (5.1d) were removed and $\mathbf{Y} = \mathbf{A}$. All other parameters for the MCMC steps are the same. The stem plots become cumbersome to read with so many comparisons, and thus a comparison matrix is given in Figures 5.7 and 5.9. For classification purposes, entries in the top two values for each comparison were taken. The tables of comparisons



(a) $\hat{\mathbf{\Phi}}$ from MDD2

(b) $\hat{\mathbf{Z}}$ from MDD2

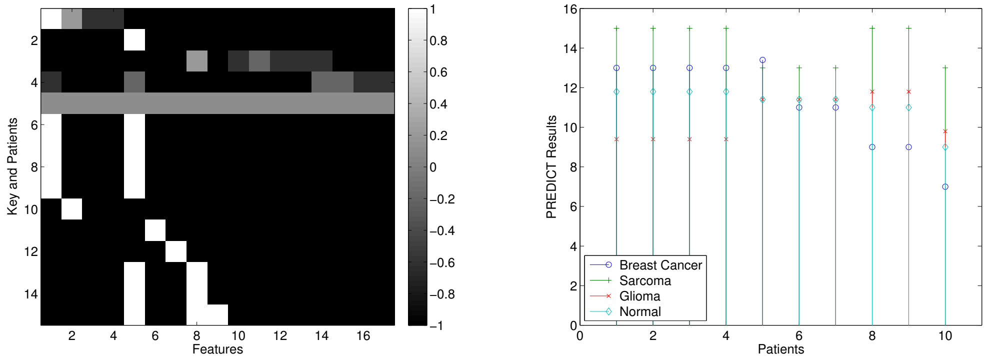
(c) $\hat{\pi}_k$ atoms in descending order from MDD2

Figure 5.5: BPFA results for MDD2 in Z-PREDICTn

for each case are given in Tables 5.5 and 5.7. This resulted in 68% combined correct classification and 32% combined error for MDD1 and 59.5% combined correct classification and 40.5% combined error for MDD2.

Key	**K1**	**K2**	**K3**	**K4**	**K5**	**R6**	**R7**	**R8**	**R9**	**R10**	**R11**	**R12**	**R13**	**R14**	**R15**
Breast Cancer (K1)	14.76	11.00	9.40	10.20	13.00	13.00	13.00	13.00	13.00	13.40	11.00	11.00	9.00	9.00	7.00
Sarcoma (K2)	11.00	17.00	11.40	13.00	15.00	15.00	15.00	15.00	15.00	13.00	13.00	13.00	15.00	15.00	13.00
Glioma (K3)	9.40	11.40	12.52	10.12	13.40	9.40	9.40	9.40	9.40	11.40	11.40	11.40	11.80	11.80	9.80
Normal (K4)	10.20	13.0	10.12	12.20	13.40	11.80	11.80	11.80	11.80	11.40	11.40	11.40	11.00	11.00	9.00

Table 5.2: Table of MDD1 Z-PREDICTn results



(a) $\mathbf{A}^T$ matrix with disease keys and patients

(b) MDD1 $\mathbf{C}$ results

Figure 5.6: MDD1 Z-PREDICTn results using all equations of (5.1)

5.3.4 Multiple Disease State Clustering Results Using Beta PDF Fitting and BPFA

The results for MDD1 are shown in Figures 5.10. For these results, 2000 burn-in iterations and 2000 sample iterations were used in the blocked Gibbs sampler with $K = 50$ and $c = 0$. The result for $\mathbf{C}$ is shown in Table 5.8 and Figure 5.12. This resulted in a combined correct classification of 65% and a combined error of 35%. For MDD2 the results may be seen in Figure 5.12 where 2000 burn-in iterations and 2000

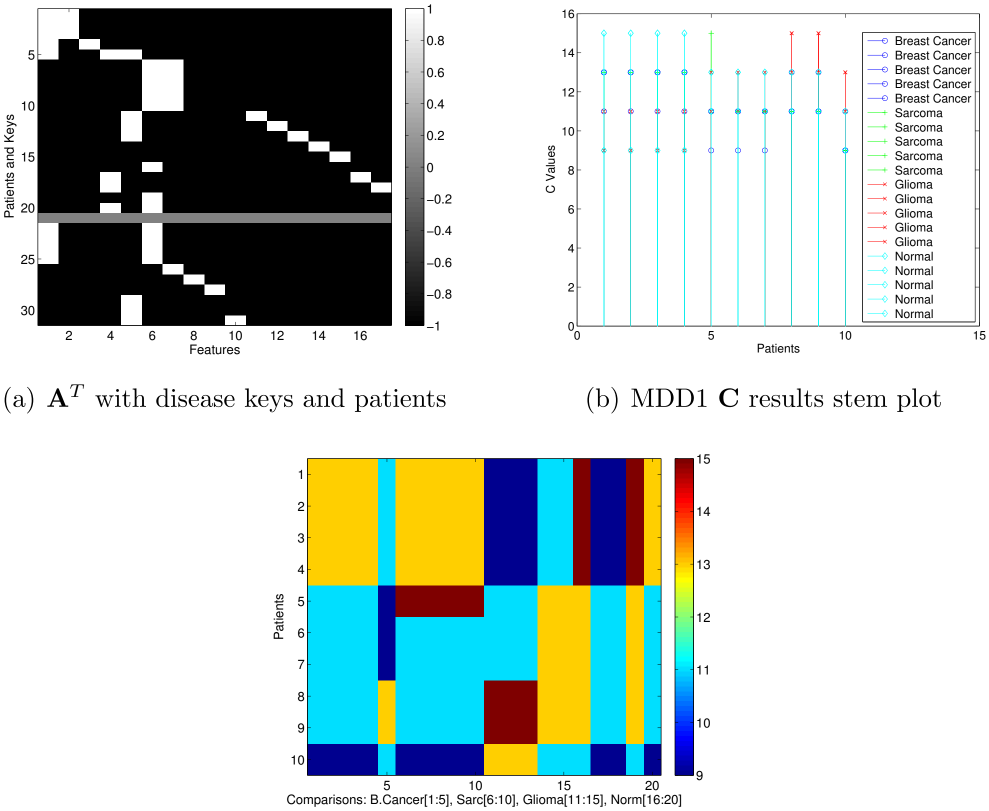
Key	**K1**	**K2**	**K3**	**K4**	**K5**	**R6**	**R7**	**R8**
Alzheimer's (K1)	14.49	13.29	14.30	12.90	15.10	15.10	15.10	15.10
Cocci (K2)	13.29	14.76	15.80	15.20	14.00	14.00	14.00	14.00
Myelmoa (K3)	14.30	15.80	19.16	17.40	15.40	15.40	15.40	15.40
Normal (K4)	12.90	15.20	17.40	20.00	14.00	14.00	14.00	14.00
Key	**R9**	**R10**	**R11**	**R12**	**R13**	**R14**	**R15**	**R16**
Alzheimer's (K1)	12.50	12.50	14.70	14.70	14.70	14.70	14.70	14.70
Cocci (K2)	12.20	12.20	14.20	14.20	14.00	14.00	14.00	14.00
Myelmoa (K3)	13.40	13.40	15.40	15.40	15.40	15.40	15.40	15.40
Normal (K4)	12.00	12.00	14.00	14.00	14.00	14.00	14.00	14.00
Key	**R17**	**R18**	**R19**	**R20**	**R21**	**R22**	**R23**	**R24**
Alzheimer's (K1)	12.30	12.30	14.90	13.90	14.90	14.90	14.90	11.90
Cocci (K2)	12.00	12.00	16.40	14.40	16.40	16.40	14.00	12.40
Myelmoa (K3)	13.40	13.40	19.40	17.40	19.40	19.40	15.40	16.60
Normal (K4)	12.00	12.00	18.00	16.00	18.00	18.00	14.00	14.00

Table 5.3: Table of MDD2 Z-PREDICTn results

Disease	**True +**	**True -**	**Type I**	**Type II**
Breast Cancer	5/5	5/5	0/5	0/5
Sarcoma	10/10	0/0	0/0	0/10
Glioma	5/5	5/5	0/5	0/5
Normal	0/0	8/10	2/10	0/0

Table 5.4: Indication and error rates for dataset 1 after Z-PREDICTn

sample iterations with $K = 50$ and $c = 100$ were used. The results for $\mathbf{C}$ may be seen in Figure 5.13 and Table 5.9. This led to a combined correct classification of 100% and a combined error of 0%.

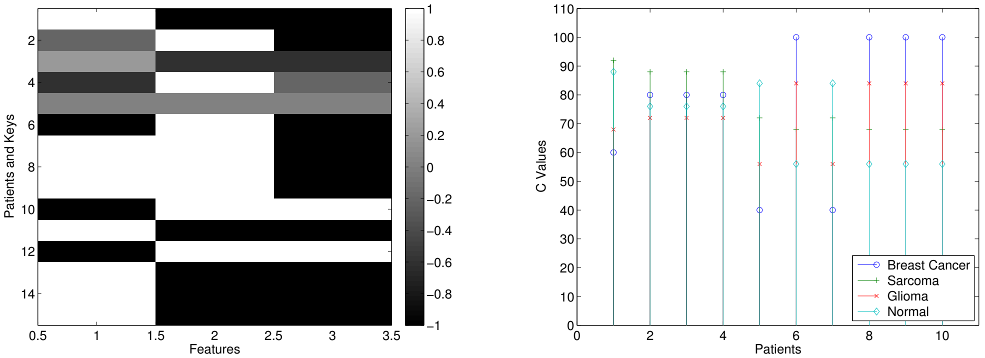


(a) $\mathbf{A}^T$ with disease keys and patients

(b) MDD1 $\mathbf{C}$ results stem plot

(c) MDD1 $\mathbf{C}$ results matrix

Figure 5.7: MDD1 Z-PREDICTn results not using (5.1c) and (d)



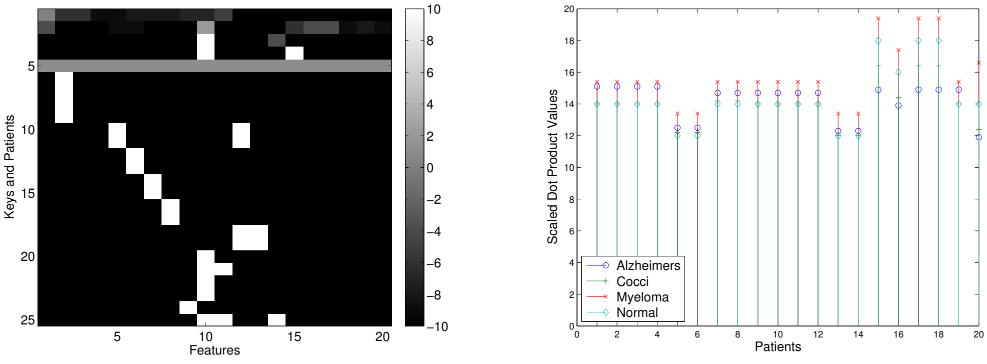
(a) $\mathbf{A}^T$ with disease keys and patients

(b) MDD1 $\mathbf{C}$ results

Figure 5.12: MDD1 results for ZBPREDICTn

Disease	**True +**	**True -**	**Type I**	**Type II**
Breast Cancer	16/25	14/25	11/25	9/25
Sarcoma	35/50	0/0	0/0	15/50
Glioma	25/25	23/25	2/25	0/25
Normal	0/0	23/50	27/50	0/0

Table 5.5: MDD1 Z-PREDICTn results with (5.1c) and (d) removed



(a) $\mathbf{A}^T$ with disease keys and patients

(b) MDD2 $\mathbf{C}$ results

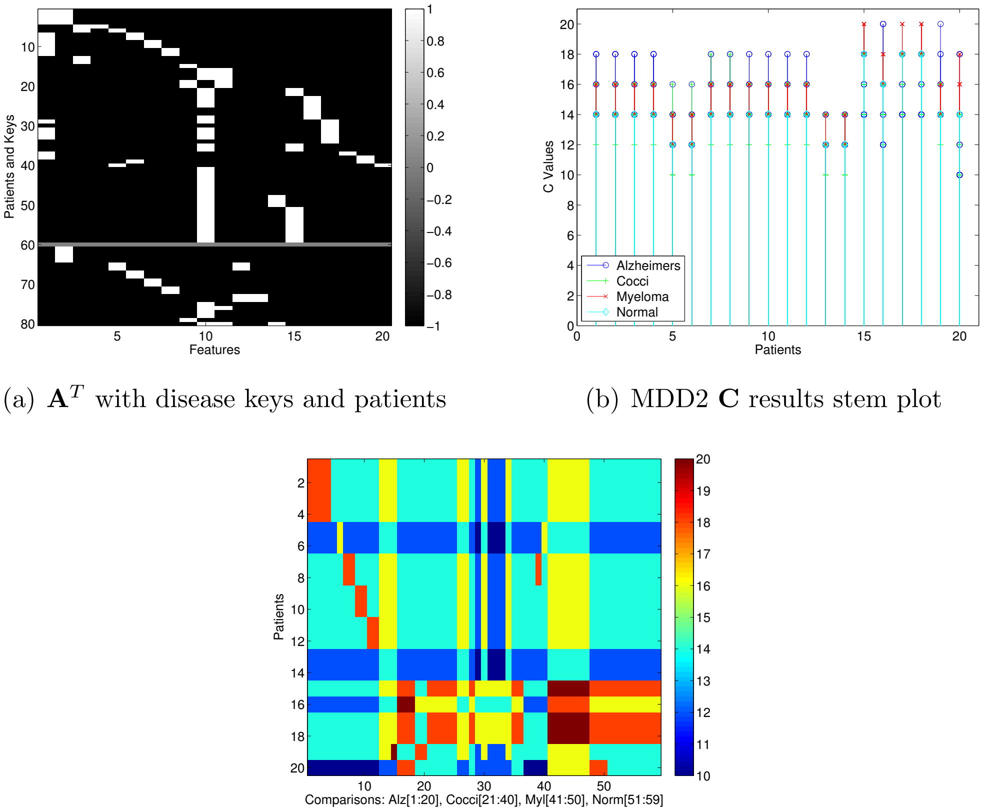
Figure 5.8: MDD2 Z-PREDICTn results

Disease	**True +**	**True -**	**Type I**	**Type II**
Alzheimer's	15/20	0/0	0/0	5/20
Cocci	0/0	20/20	0/20	0/0
Myeloma	20/20	0/0	0/0	0/20
Normal	0/0	15/20	5/20	0/0

Table 5.6: MDD2 Z-PREDICTn results.

Disease	**True +**	**True -**	**Type I**	**Type II**
Alzheimer's	124/400	0/0	0/0	276/400
Cocci	0/0	292/400	108/400	0/0
Myeloma	151/200	0/0	0/0	49/200
Normal	0/0	135/180	45/180	0/0

Table 5.7: MDD2 Z-PREDICTn results with (5.1c) and (d) removed



(a) $\mathbf{A}^T$ with disease keys and patients

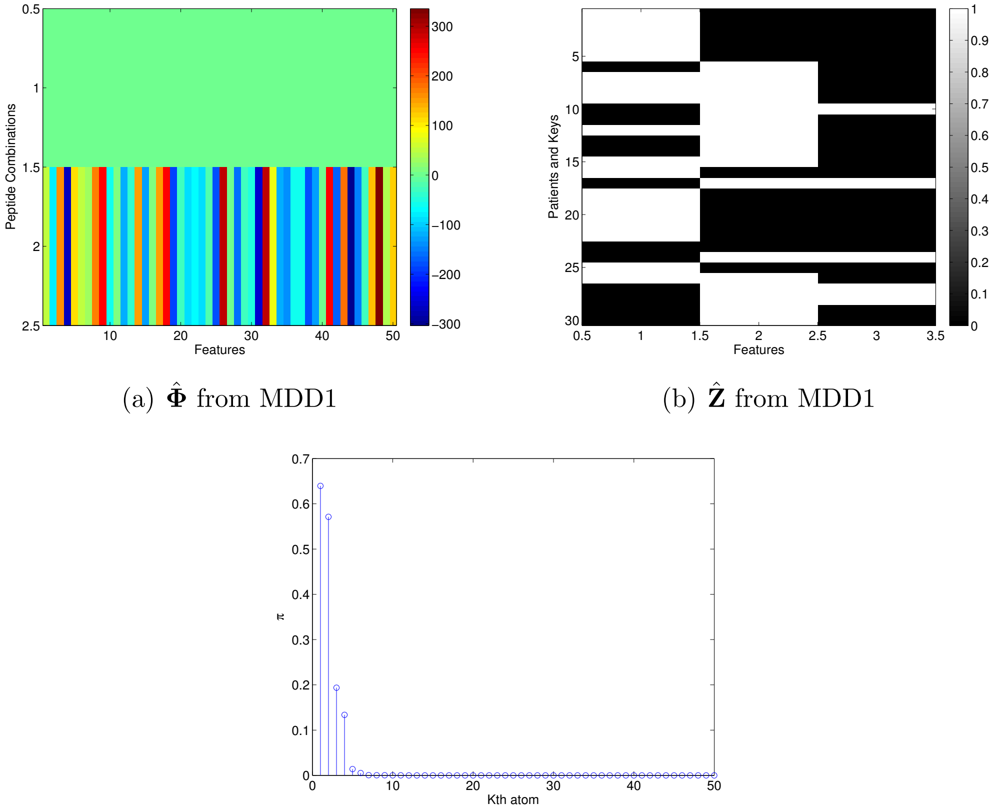
(b) MDD2 **C** results stem plot

(c) MDD2 **C** results matrix

Figure 5.9: MDD2 Z-PREDICTn results not using (5.1c) and (d)

Key	**K1**	**K2**	**K3**	**K4**	**R6**	**R7**	**R8**
Breast Cancer (K1)	30.00	-2.00	14.00	-14.00	-10.00	10.00	10.00
Sarcoma (K2)	-2.00	20.40	-0.40	13.20	22.00	18.00	18.00
Glioma (K3)	14.00	-0.40	7.60	-6.00	-2.00	2.00	2.00
Normal (K4)	-14.00	13.20	-6.00	14.00	18.00	6.00	6.00
Key	**R9**	**R10**	**R11**	**R12**	**R13**	**R14**	**R15**
Breast Cancer (K1)	10.00	-30.00	30.00	-30.00	30.00	30.00	30.00
Sarcoma (K2)	18.00	2.00	-2.00	2.00	-2.00	-2.00	-2.00
Glioma (K3)	2.00	-14.00	14.00	-14.00	14.00	14.00	14.00
Normal (K4)	6.00	14.00	-14.00	14.00	-14.00	-14.00	-14.00

Table 5.8: ZBPREDICTn **C** for MDD1

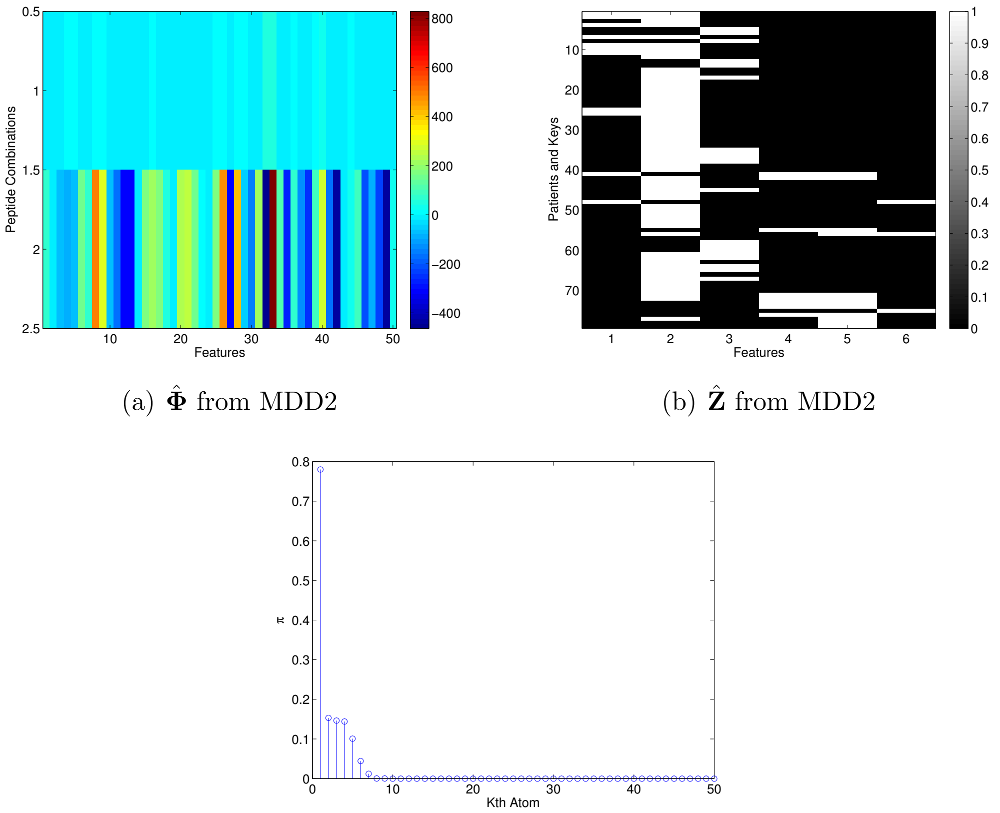


(a) $\hat{\mathbf{\Phi}}$ from MDD1

(b) $\hat{\mathbf{Z}}$ from MDD1

(c) $\hat{\pi}_k$ atoms in descending order from MDD1

Figure 5.10: ZBPREDICTn results for MDD1

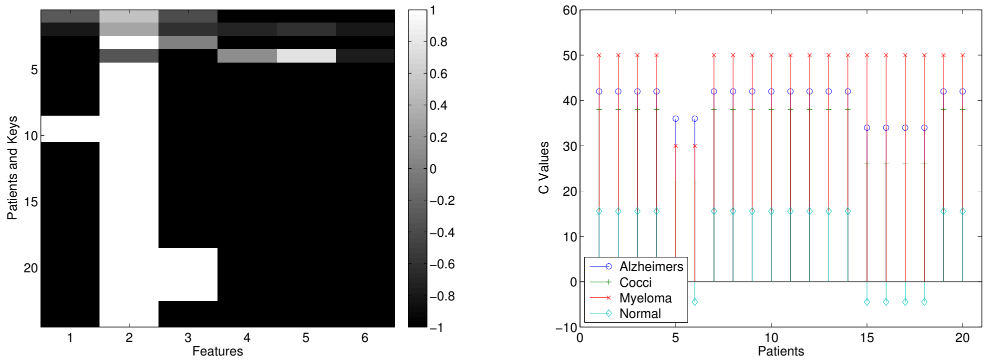


(a) $\hat{\mathbf{\Phi}}$ from MDD2

(b) $\hat{\mathbf{Z}}$ from MDD2

(c) $\hat{\pi}_k$ atoms in descending order from MDD2

Figure 5.11: BPFA results for MDD2



(a) $\mathbf{A}^T$ with disease keys and patients

(b) MDD2 $\mathbf{C}$ results

Figure 5.13: MDD2 results for ZB-PREDICTn

Key	K1	K2	K3	K4	K5	R6	R7	R8
Alzheimer's (K1)	35.00	27.30	38.00	4.22	42.00	42.00	42.00	42.00
Cocci (K2)	27.30	25.80	32	13.77	38.00	38.00	38.00	38.00
Myelmoa (K3)	38.00	32.00	50.00	5.55	50.00	50.00	50.00	50.00
Normal (K4)	4.22	13.77	5.55	33.33	15.55	15.55	15.55	15.55

Key	R9	R10	R11	R12	R13	R14	R15	R16
Alzheimer's (K1)	36.00	36.00	42.00	42.00	42.00	42.00	42.00	42.00
Cocci (K2)	22.00	22.00	38.00	38.00	38.00	38.00	38.00	38.00
Myelmoa (K3)	30.00	30.00	50.00	50.00	50.00	50.00	50.00	50.00
Normal (K4)	-4.44	-4.44	15.55	15.55	15.55	15.55	15.55	15.55

Key	R17	R18	R19	R20	R21	R22	R23	R24
Alzheimer's (K1)	42.00	42.00	34.00	34.00	34.00	34.00	42.00	42.00
Cocci (K2)	38.00	38.00	26.00	26.00	26.00	26.00	38.00	38.00
Myelmoa (K3)	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00
Normal (K4)	15.55	15.55	-4.44	-4.44	-4.44	-4.44	15.55	15.55

Table 5.9: Table of MDD2 ZBPREDICTn results

Disease	True +	True -	Type I	Type II
Breast Cancer	3/5	1/5	4/5	2/5
Sarcoma	6/10	0/0	0/0	4/10
Glioma	4/5	5/5	0/5	1/5
Normal	0/0	7/10	3/10	0/0

Table 5.10: Indication and error rates for MDD1 after ZB-PREDICTn

Disease	True +	True -	Type I	Type II
Alzheimer's	20/20	0/0	0/0	0/20
Cocci	0/0	20/20	0/20	0/0
Myeloma	20/20	0/0	0/0	0/20
Normal	0/0	20/20	0/20	0/0

Table 5.11: Indication and error rates for MDD2 after ZB-PREDICTn

Chapter 6

H-PREDICT AND HB-PREDICT FOR MULTIPLE STATES

While Dirichlet process Gaussian mixture modeling (DP-GMM) is useful for classification in applications where there are multiple unknown states that need to be adaptively clustered, it is not able to provide classifications in more than one group for a single patient. One way around this issue is to use the hierarchical Dirichlet process (HDP) in conjunction with modified beta process factor analysis (BPFA), which simply places a prior over the initial distribution, and then executes the DP-GMM as given in Algorithm 1. While it is possible to apply patient comparisons as inputs to the DP-GMM, further bookkeeping is required. This is simplified with the HDP.

6.1 Hierarchical Dirichlet Process Theory

The HDP is simply an extension of the DP to include an additional DP prior on the base measure. In the same way that the DP may be thought of as a "Chinese restaurant process" (CRP), the HDP may be thought of as a "Chinese restaurant franchise" (CRF) [103], or a non-parametric extension of linear Dirichlet allocation [104]. The HDP is very flexible in that it allows for multiple hierarchies, which is useful in diverse topics such as music, target tracking, and speech [105–108]. Assuming a base distribution S, the hierarchical extension is given by:

$$\begin{aligned} G_0 &= \sum_{m=1}^{\infty} \beta_m \delta(\theta - \theta_{h,m}) && (6.1a) \\ G_j &= \sum_{m=1}^{\infty} \pi_{j,m} \delta(\theta - \theta_{h,m}) && (6.1b) \\ \theta_{h,m} &\sim S && (6.1c) \end{aligned}$$

An extension to Equation (2.20, where $j = 1 \ldots J$ is the cohort number, forms a stick-breaking representation of the HDP [109]:

$$\begin{aligned}
G_0|\gamma_h, A &\sim \mathrm{DP}(\gamma_h, A), & (6.2a)\\
G_j|\alpha_h, G_0 &\sim \mathrm{DP}(\alpha_h, G_0), & (6.2b)\\
\beta'_k &\sim \mathrm{Beta}(1, \gamma_h),\ \ k = 1, \ldots, \infty, & (6.2c)\\
\beta_m &= \beta'_k \prod_{l=1}^{m-1} (1 - \beta'_l),\ \ m = 1, \ldots, \infty, & (6.2d)\\
\pi_{j,k}^{(h')} &\sim \mathrm{Beta}(\alpha_h \beta_m, \alpha_h (1 - \sum_{k=1}^{m} (\beta_k))),\ \ k = 1, \ldots, \infty, & (6.2e)\\
\pi_{j,k}^{(h)} &= \pi_{j,k}^{(h')} \prod_{l=1}^{m-1} (1 - \pi_{j,l}^{(h')}),\ \ m = 1, \ldots, \infty & (6.2f)
\end{aligned}$$

The HDP equations are identical to the DP, except now a prior has been placed on the base distribution G_j. This prior for the new base distribution is itself a Dirichlet process with an innovation parameter γ_h and an underlying distribution S. Note that α_h is an innovation parameter in the DP. The subscript h in Equation (6.2) denote the HDP dependence. The conditional distribution over which an item (patient) may belong to a particular group (disease cluster) is given by [103, 109]:

$$m_{j,i}|m_{j,1:i-1}, \alpha_h \sim \sum_m \frac{n_{j,m}}{\sum_{m'} n_{j,m'} + \alpha_h} \delta(\theta - \theta_{h,m}) + \frac{\alpha_h}{\sum_{m'} n_{j,m'} + \alpha_h} \delta(\theta - \theta_{h,m}^{new}) \quad (6.3)$$

where m is a particular cluster and $n_{j,m}$ is the number of patients already present in a cluster. While only two hierarchical levels are explored here, the HDP can be extended to as many as desired depending on the problem to be solved. The blocked Gibbs sampler algorithm with update equations for may be found in Algorithm 3.

Several papers have focused on using the HDP as a remedy to the co-clustering problem, including [110] and [111] for haplotype reconstruction where data is unlikely to be pooled together. It is also investigated in [112] in a multilevel format for human EEG monitoring and in [113] for brain fiber tract clustering. In each of these cases, some modification of the HDP or data set model was required in order to successfully perform the co-clustering. The immunosignaturing results determined in Z-PREDICTn and ZB-PREDICTn lend themselves to direct implementation of the HDP, as the data has a natural separation into groups per patient. Note that while it is possible to apply the DP-GMM as well to the results of the Z-PREDICTn and ZB-PREDICTn, additional bookkeeping would need to be performed in order to first combine all patient results into a single group and then further to break the resulting clustering into groups per patient. While the DP-GMM itself may be relatively straightforward for cohorts with few patients present, it becomes more tedious if the patient numbers in each cohort increases.

Although work has been done to make the implementation of the HDP more efficient [114], implementation via a blocked Gibbs sampler is investigated here to maintain consistency with prior immunosignaturing work performed with the DP-GMM. In the case of immunosignaturing, two levels will be sufficient to allow for clustering across multiple cohorts, each with their own patient groups. This is because the first level of clustering (local) will take care of the group assignment to a particular disease state while the second level (global) will ensure that the available clusters will be the same across all available cohorts.

6.2 HDP Based Multiple Disease State Clustering

6.2.1 PCA Features with HDP

While successful clustering for immunosignaturing was previously demonstrated for single diseases using PREDICT, B-PREDICT, BIO-PREDICT, and BIOB-PREDICT, it did not extend the clusters to disease diagnosis or multiple disease states. While this may be sufficient if only the total number of disease states is desired, or if one is only interested in the incidence of a new disease in a population, missing diagnoses is impractical at a clinical level for everyday medicine. Therefore, it is desirable to extend the previous results to include diagnosis information in addition to the ability to extend beyond a single disease diagnosis.

The HDP is a natural addition to this work in that generally diagnosis requires some sort of training data where the ground truth disease state is known. As such, a method is devised that builds upon the Z-PREDICTn previously discussed in that the results of **C** are then fed into the HDP for adaptive clustering. This method is called H-PREDICT, or **H**DP of **P**CA **RE**solution and **D**iscovery for **I**mmunosignature **C**lassification **T**esting.

6.2.2 Beta PDF Fitting Features with HDP

Similar to how PCA can be used to reduce the dimensionality in the H-PREDICT algorithm, BPDFF can be used as an alternative. The output of ZB-PREDICTn may be fed into the HDP. This is called HB-PREDICT, or **H**DP of **B**eta **P**DF **RE**solution and **D**iscovery for **I**mmunosignature **C**lassification **T**esting. This allows the user to restrict the dimensionality to two, rather than an indeterminate number of principal components.

6.3 HDP Clustering Results

6.3.1 Clustering Results of PCA with HDP

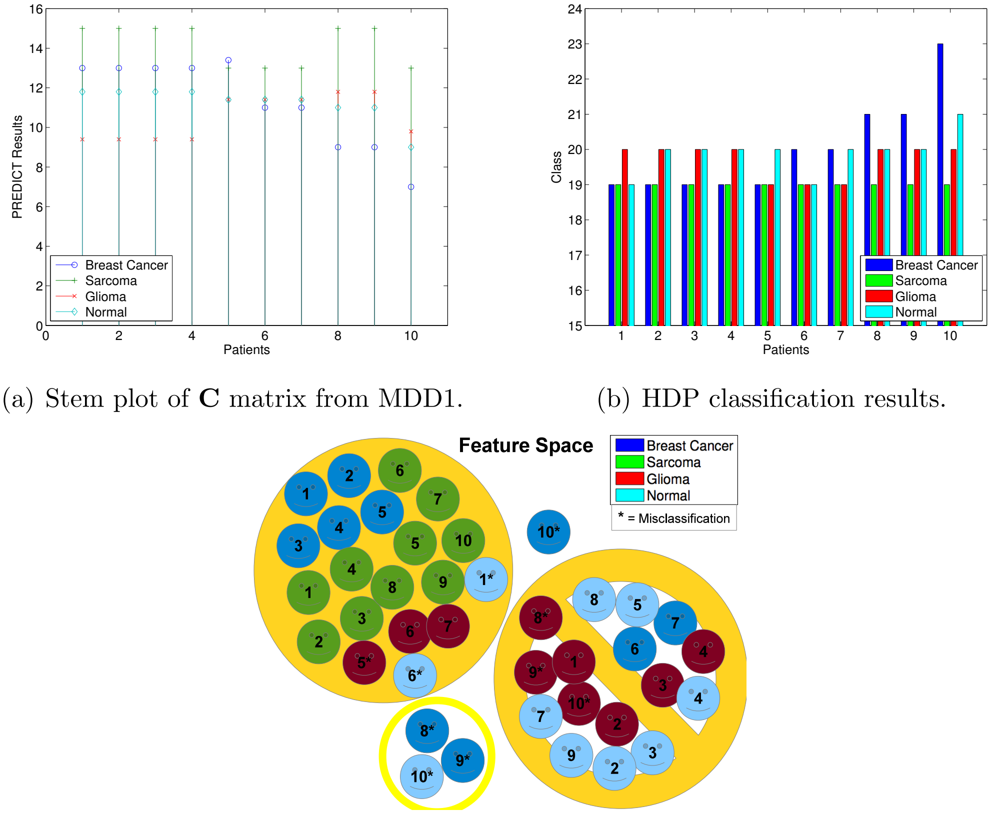
The H-PREDICT algorithm was tested against with datasets MDD1 and MDD2. The MDD1 classification results corresponding to Figure 5.6 are given in

Figure 6.1. The **C** matrix stem plot is also given again for reference. The classification results are also summarized in an additional graphical representation as well. In this depiction, each face has a number associated with it that corresponds to which patient the data point originates from. Each group represents a cluster created in H-PREDICT. The clusters correspond to having a state present (solid circle) while the disease not present cluster has a slash through the cluster. Entries that are incorrectly clustered have an asterix (*) next to their patient number. Table 6.1 summarizes the classification results, with a combined correct classification of 75% and a combined error rate of 12.5%. The innovation parameters were $\alpha_h = \gamma_h = 3$ with 5000 burn-in iterations and 5000 sample iterations.

The MDD2 classification results corresponding to Figure 5.8 are given in Figure 6.2. The **C** matrix stem plot is also given again for reference. The correct combined classification was 57.5% while the combined error rate was 20%. The innovation parameters were $\alpha_h = \gamma_h = 6$ and for the blocked Gibbs sampler, 2000 burn-in iterations and 5000 sample iterations were used.

Disease	**True +**	**True -**	**Type I**	**Type II**	**No Result**
Breast Cancer	5/5	2/5	0/5	0/5	3/10
Sarcoma	10/10	0/0	0/0	0/10	0/10
Glioma	2/5	4/5	1/5	3/5	0/10
Normal	0/0	7/10	2/10	0/0	1/10

Table 6.1: MDD1 H-PREDICT classification summary



(a) Stem plot of **C** matrix from MDD1.

(b) HDP classification results.

(c) HDP classification results.

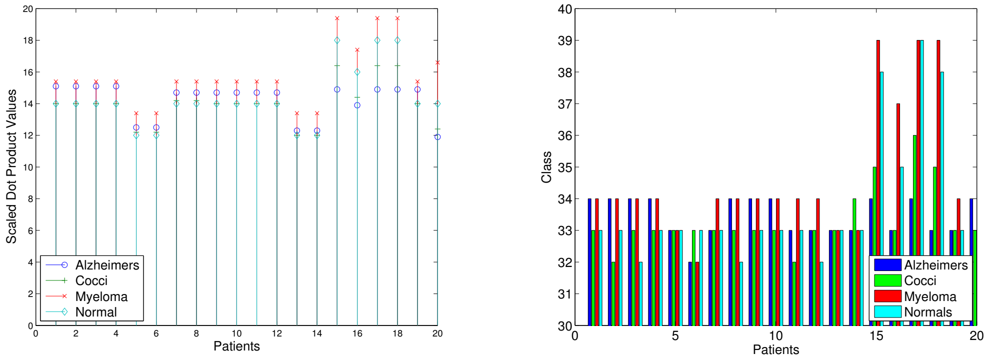
Figure 6.1: H-PREDICT results for MDD1

Disease	**True +**	**True -**	**Type I**	**Type II**	**No Result**
Alzheimers	10/20	0/0	0/0	9/20	1/20
Cocci	0/0	14/20	1/20	0/0	5/20
Myeloma	11/20	0/0	0/0	3/20	6/20
Normal	0/0	13/20	0/20	0/0	7/20

Table 6.2: MDD2 H-PREDICT classification summary

6.3.2 Clustering Results of Beta PDF Fitting with HDP

The HB-PREDICT classification results for MDD1 are given in Figure 6.3. The stem plot of the **C** matrix is also given again for reference. The innovation

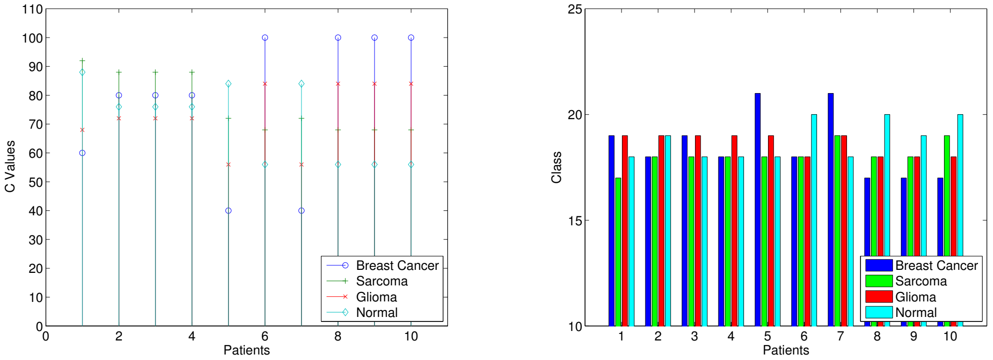


(a) Modified $\hat{\mathbf{Z}}$ matrix from reduced MDD2.

(b) HDP classification results.

Figure 6.2: H-PREDICT results for MDD2

parameters were $\alpha_h = \gamma_h = 4$ and 2000 burn-in followed by 2000 sample iterations were used. The combined correct classification was 50% while the combined error was 25%. The HB-PREDICT MDD2 classification results are given in Figure 6.4 with the classification results summarized in Table 6.4. The combined correct classification was 70% while the combined error was 15%.



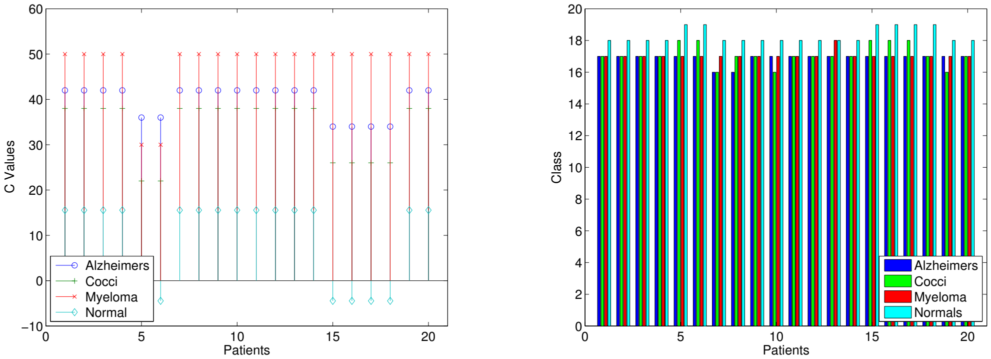
(a) Modified $\hat{\mathbf{Z}}$ matrix from MDD1.

(b) HDP classification results.

Figure 6.3: HB-PREDICT results for MDD1

Disease	**True +**	**True -**	**Type I**	**Type II**	**No Result**
Breast Cancer	2/5	0/5	1/5	2/5	5/10
Sarcoma	7/10	0/0	0/0	2/10	1/10
Glioma	4/5	5/5	0/5	1/5	0/10
Normal	0/0	2/10	5/10	0/0	3/10

Table 6.3: MDD1 HB-PREDICT classification summary



(a) Modified $\hat{\mathbf{Z}}$ matrix from MDD2.

(b) HDP classification results.

Figure 6.4: HB-PREDICT results for MDD2

Disease	**True +**	**True -**	**Type I**	**Type II**	**No Result**
Alzheimers	18/20	0/0	0/0	2/20	0/20
Cocci	0/0	5/20	12/20	0/0	3/20
Myeloma	19/20	0/0	0/0	1/20	0/20
Normal	0/0	14/20	0/20	0/0	6/20

Table 6.4: MDD2 HB-PREDICT classification summary

6.4 H-PREDICT and HB-PREDICT Model Challenges

There are several difficulties with using the HDP in a classification framework as presented in the previous sections. First, with both algorithms, a two cluster approach is used to indicate which diseases are present. It is possible that there are multiple clusters of interest to the user, rather than just a binary disease diagnosis.

Algorithm 3 Blocked Gibbs sampling for HDP using an D-dimensional cohort $\mathbf{X}$ with J individual subgroups

Input: J datasets $\mathbf{X} = \{\mathbf{x}_1 \ldots, \mathbf{x}_N\}$, HDP innovation parameters α_h and γ_h, Normal-Wishart hyperparameters $\boldsymbol{\mu}_{\boldsymbol{\mathcal{N}}}, \tau_{\mathcal{N}}, \xi_{\mathcal{W}}, \iota_{\mathcal{W}}$, DP truncation limit M.

Output: Samples $\{\boldsymbol{\mu}_m^{(i)}, \boldsymbol{\Sigma}_m^{(i)}, \mathbf{c}^{0(i)}, \mathbf{w}_{0,\mathbf{J}}^{(i)}\}_{i=1}^{L}$

Repeat for $i = 1, 2, \ldots,$ Gibbs iterations:

1. Update for $\boldsymbol{\theta}_m^{(i)} = \{\boldsymbol{\mu}_m^{(i)}, \boldsymbol{\Sigma}_m^{-1(i)}\} \sim p\,(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m^{-1} | \mathbf{c}^{(i-1)}, \mathbf{X})$, $m = 1, \ldots, M$.
   - (a) Let $\mathbf{X}_m = \{\mathbf{x}_n : c_n^{(i-1)} = m\}$ and $N_m = |\mathbf{X}_m|$, for $m = 1, \ldots, M$.
   - (b) For all clusters, $m = 1, \ldots, M$, compute,

$$
\begin{aligned}
\boldsymbol{\mu}_{\mathbf{x}_m} &= \frac{1}{N_m} \sum_{n : c_n^{(i-1)} = m} \mathbf{x}_n \\
\boldsymbol{\Sigma}_{\mathbf{x}_m} &= \frac{1}{N_m} \sum_{n : c_n^{(i-1)} = m} (\mathbf{x}_m - \boldsymbol{\mu}_{\mathbf{x}_m})^2 \\
\tilde{\boldsymbol{\mu}}_{\mathcal{N},m} &= \frac{\tau_{\mathcal{N}}\, \tilde{\boldsymbol{\mu}}_{\mathcal{N}} + N_m\, \boldsymbol{\mu}_{\mathbf{x}_m}}{\tau_{\mathcal{N}} + N_m}\,, \\
\tilde{\tau}_{\mathcal{N},m} &= \tau_{\mathcal{N}} + N_m\,, \\
\tilde{\iota}_{\mathcal{W},m} &= \iota_{\mathcal{W}} + \boldsymbol{\Sigma}_{\mathbf{x}_m} + \frac{\tau_{\mathcal{N}}\, N_m}{\tau_{\mathcal{N}} + N_m} \left(\mathbf{m} - \boldsymbol{\mu}_{\mathbf{x}_m}\right) \left(\mathbf{m} - \boldsymbol{\mu}_{\mathbf{x}_m}\right)^T, \\
\tilde{\xi}_{\mathcal{W},m} &= \xi_{\mathcal{W}} + N_m\,.
\end{aligned}
$$

   - (c) Draw samples for $\boldsymbol{\Sigma}_m^{-1(i)}$ from the Wishart distribution, $\mathcal{W}\,(\boldsymbol{\Sigma}_m^{-1}; \tilde{\iota}_{\mathcal{W},m}, \tilde{\xi}_{\mathcal{W},m})$, for $m = 1, \ldots, M$.
   - (d) Finally draw samples for $\boldsymbol{\mu}_m^{(i)}$ from the Normal distribution, $\mathcal{N}\,(\boldsymbol{\mu}_m; \tilde{\boldsymbol{\mu}}_{\mathcal{N},m}, \frac{\boldsymbol{\Sigma}_m^{(i)}}{\tilde{\tau}_{\mathcal{N},m}})$, for $m = 1, \ldots, M$.
2. Do the global update.
   - (a) Draw samples $\beta_g \sim \mathrm{Beta}\Big(1 + N_m^*, \gamma_h + \sum_{m'=m+1}^{M} N_{m'}^*\Big)$,
     where $N_m^* \triangleq |\{n : c_n^{(i)} = m\}|$, $m = 1, \ldots, M$.
   - (b) Update for $w_m^{(i)} \sim p\,(w_m^0 | \mathbf{c}^{(i)})$, $m = 1, \ldots, M$.
     Finally evaluate $w_m^{(0,i)} = \beta_m \prod_{g=1}^{m-1} (1 - \beta_g)$, $m = 1, \ldots, M$.

Algorithm 3 Continued

```
3) Do the local updates for all $J$ groups
    a) Let $q^j_{m,n} \triangleq w^{j,(i-1)}_m \mathcal{N}(\mathbf{x}^j_n; \boldsymbol{\mu}^{(i)}_m, \boldsymbol{\Sigma}^{(i)}_m)$, $m = 1, \ldots, M$ and $n = 1, \ldots, N$.
    b) Normalize $q^{j,\prime}_{m,n} = \frac{q^j_{m,n}}{\sum_{m=1}^{M} q^j_{m,n}}$, $m = 1, \ldots, M$ and $n = 1, \ldots, N$.
    c) Update for $c^{j,(i)}_n \sim p(c^j_n | \boldsymbol{\mu}^{(i)}, \boldsymbol{\Sigma}^{-1(i)}, \mathbf{w}^{(i-1)}, \mathbf{X})$, $n = 1, \ldots, N$. Draw samples for $c^{j,(i)}_n \sim \sum_{m=1}^{M} q^{j,\prime}_{m,n} \delta(c^j_n, m)$, $n = 1, \ldots, N$. Note that $\mathbf{c} = c^{1:J}$.
    d) Draw samples $\beta^j_l \sim \text{Beta}\Big(\alpha_h w^{j,(i-1)} + N^*_m, \alpha_h(1 - w^{j,(i-1)})\Big)$, where $N^*_m \triangleq |\{n : c^{j,(i)}_n = m\}|$, $m = 1, \ldots, M$.
    e) Update for $w^{j,(i)}_m \sim p(w^j_m | \mathbf{c}^{j,(i)})$, $m = 1, \ldots, M$. Finally evaluate $w^{j,(i)}_m = \beta^j_l \prod_{l=1}^{m-1} (1 - \beta_l)$, $m = 1, \ldots, M$.
```

As such, additional disease or patient information would be required in order to interpret additional clusters. Further, when ”no result” groups are created, additional information may be required in order to determine patient diagnosis. Fortunately, because this method is executed in a multidimensional embodiment, it is possible to add other patient features or data points as a multidimensional patient vector, and classify hierarchically based on that information as well. Additionally, with every additional layer of hierarchy, computational complexity increases, leading to increased execution time for convergence in the blocked Gibbs sampler.

Chapter 7

COMPARISON OF PROPOSED CLUSTERING ALGORITHMS

Now that various methods have been proposed and applied to various datasets, the results discussion is presented. In the case of the single disease methods, this includes comparison with a known method that performs well for immunosignaturing analysis. For the multi-disease methods, no alternative analysis has been presented for immunosignaturing, and alternate Bayesian non-parametrics are not suitable without further modification. Each method's link to the biological model is discussed in order to provide a background and reasoning for some of the model selection.

7.1 Method relationships to the biologic model

A variety of methods have been presented, each with their own benefits and tradeoffs for immunosignaturing analysis. Before these results are discussed in depth, it is helpful to have an understanding of the biological models that they each represent. As with most models, these seek to represent the natural phenomena (in this case disease pathology) in a simplified way for better understanding. In terms of immunosignaturing arrays, it has been shown that the presence of a disease state will be indicated by the binding of antibodies specific to that particular pathology to the random, but known, peptide sequences present on the glass immunosignaturing slide. Particular antibodies will bind with varying affinities to the peptide sequences present in the spot locations based on chemical interactions [15]. As such, the fluorescence combinations will be indicative of various disease states [7–13].

In the case of PREDICT (PCA REsolution with DP-GMM for Immunosignature Classification Testing) and B-PREDICT (Beta PDF REsolution with DP-GMM for Immunosignature Classification Testing), these combinations are reduced to either distribution parameters that represent the binding behavior (B-PREDICT) or principal component analysis (PCA) components used to eliminate small binding contributions (PREDICT), and then the Dirichlet process Gaussian mixture model (DP-GMM) is used to classify patients based on these parameters. In the case of PREDICT, the PCA components themselves are representative of combinations of peptide binding results, and thereby are the high intensity value spots of interest. Since these are discernible by data inspection, these are considered visible processes to the viewer. While there is still useful information to be gleaned from the peptides that are not high intensity, because the peptide array is fixed, if one knows which peptides are considered high intensity, then it is also known which are not considered high intensity.

It should also be noted that the threshold for significant peptide combinations is determined by the eigenvalue analysis, which is user determined. Thus, the user is indirectly setting a threshold for intensity significance. For B-PREDICT rather than directly or indirectly imposing a significance threshold for the intensity measurements, a more holistic view is taken to describe the distribution of all the peptide intensity measurements. This has some advantages in that no peptides are discarded in the analysis, aside from those having no or negative measurements from the actual equipment, which are discarded in in any case. Interestingly, the PCA and beta PDF fitting (BPF) dimensionality reductions are complementary, and datasets may do better with one or the other based on their antibody characteristics. For example, it is possible that peptides of interest will be discarded during the eigenvalue analysis, especially if only a few small pathological changes are present. These changes may be lost due to the imposed thresholding, and important disease information subtleties will be lost. Conversely, the BPF version will miss multi-modal distributions and their subtle changes, and may not have the resolution to distinguish distributions that are very similar, but whose underlying peptide combinations are very different. As such, it is important to retain both dimensionally reduction techniques and allow the user to apply the case more suited to their analysis, even though this may require additional data or user consideration.

In the case of Φ-PREDICT (**Φ** PCA REsolution and Discovery for Immunosignature Classification Testing), Z-PREDICT (Z-matrix from PCA REsolution and Discovery for Immunosignature Classification Testing), and ZB-PREDICT (Z-matrix from Beta PDF REsolution andDiscovery for Immunosignature Classification Testing), the beta process factor analysis (BPFA) is performed on the BPF or PCA down-selected dataset (i.e. encoding of the discernible feature space). In this way, latent features are discovered for each algorithm. As previously discussed, preferential antibody binding leads to high intensity spot combinations, which are then summarized through BPF or PCA and then classified to indicate disease pathology. When the BPFA is introduced, it creates an additional step in the process where by the disease pathology is now linked to K latent features, and those K latent features are linked both to patient and to the high intensity peptides of interest. As such, the classification is performed on the latent features themselves, rather than the metrics that are directly related to a biological phenomena. It should also be noted that while the BPFA can show the latent feature relationships, it does not identify the latent features themselves. In the case of immunosignaturing, this means that the algorithm does not provide a link to a biological phenomena for the K features. However, this fact itself does not prevent clustering, as in BIO-PREDICT (BPFA Including prOcessing with PCA REsolution and DP-GMM for Immunosignature Classification Testing) or BIOB-PREDICT (BPFA Including prOcessing with Beta PDF REsolution with DP-GMM for Immunosignature Classification Testing), or further analysis, such as in the case where peptide analysis is studied. Peptide analysis was not the focus of this work, but as each of these methods was able to identify patients with similar feature profiles, the BPFA may also be used to study peptides of significance for each disease. It is interesting to note that in order to toggle between methods that are geared towards examining disease diagnosis and those that are geared towards peptide analysis, all one needs to do is transpose the initial data matrix X containing the median peptide intensities, and then execute the proper method. This will similarly change the dimensionality from peptides (in the case where disease diagnosis is the preferred path) to patients (in the case where peptide analysis is desired). These methods allow for complementary analysis to occur when using the same dataset, and represent diverse options in the toolbox for those studying immunosignaturing.

When considering multiple diseases, the same biological ties still exist, but they are simply expanded to include a wider range of disease possibilities. In Z-PREDICTn (Z-matrix from PCA REsolution and Discovery for Immunosignature Classification Testing up to n-states) and ZB-PREDICTn (Z-matrix from Beta PDF REsolution and Discovery for Immunosignature Classification Testing up to n-states), the subsequent Z matrix analyses are done simply to compare known disease profiles of the patients. Thus, the same latent K features are now known for some diseases (although they are still not related to a known biological phenomena), and these can be compared to the K feature profiles for patients with unknown disease pathologies. While these methods are demonstrated in the presence of multiple diseases for each patient, they are still valid in single disease settings, especially when there is a desire to understand disease relationships. It is possible that disease pathogens that share biological similarities will share very similar pathology responses. One example of this would be the relationship of small pox to that of cow pox. Although these are separate pathogens that affect different species, their relationships to one another have allowed for similar antibody responses from the human immune system [42]. Thus, if diseases share similar combinations of high intensity spots, it may be possible that they share similar immunosignaturing intensity distribution parameters or similar high intensity peptide combinations. As such, it is possible that they will share similar latent features, thereby leading to similar Z matrices, and similar clustering results. Thus, these methods may be useful to show disease similarities, such as structure or phylogenic relationships. Further work may be done in this area to better explore disease relationships.

It is also useful to note that in Φ-PREDICT, Z-PREDICT, ZB-PREDICT, BIO-PREDICT, BIOB-PREDICT, Z-PREDICTn, and ZB-PREDICTn, the unweighted BPFA algorithm is used. Though more restrictive than the weighted case, this was thought to be the simplest case from the biological model standpoint in that each estimated matrix could be used to describe a particular biological relationship. For example, in Z-PREDICT, the Z matrix indicates the relationship between latent features and patients while the Φ matrix represents the relationship between the latent features and prominent peptide intensity combinations. The introduction of a matrix of weights would require further biological understanding for which no data was taken for immunosignaturing. However, this leads to some assumptions with the biological model being represented, especially in the case of multiple diseases. It is possible that antibodies may bind to multiple peptide spots and that there may be some competition between antibodies for binding sites [19]. The biological model in this case assumes that this is not a dominant affect. It is possible that the introduction of the weight matrix in the BPFA may indicate which peptide combinations are at odds with one another when binding (outside the scope of this work).

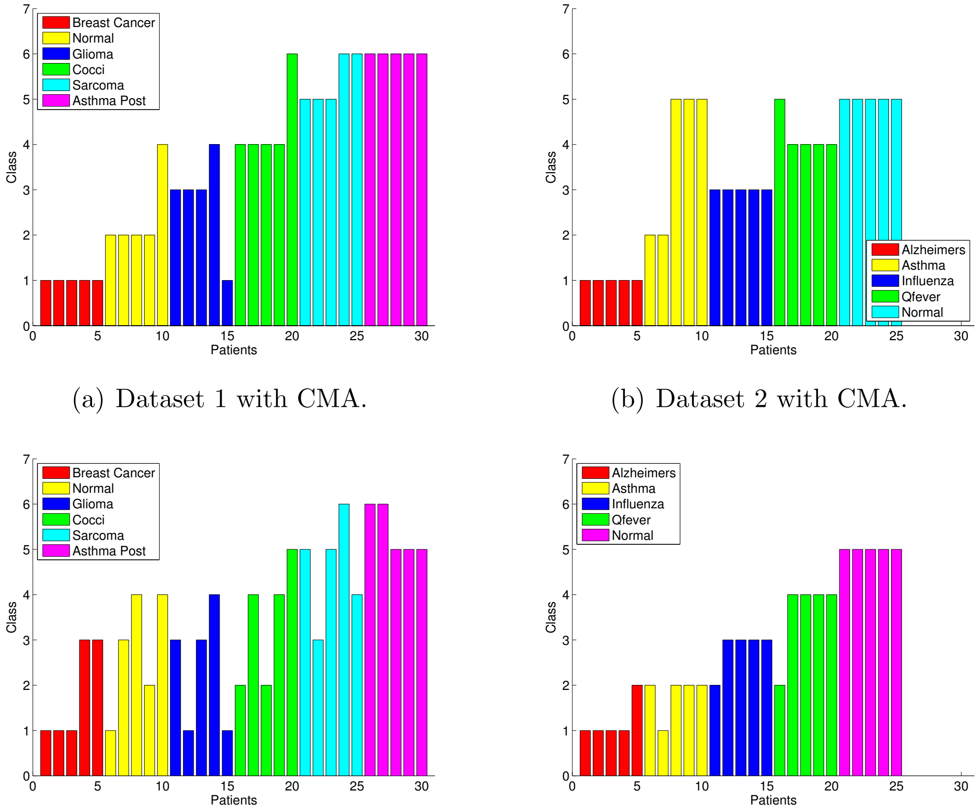
Thus, as the relationships become more complicated, an adaptive classification scheme is useful. While it is possible to use the DP-GMM directly on the

Z-PREDICTn or ZB-PREDICTn results, this would lead to additional bookkeeping. As such, the HDP is a useful tool yielding the H-PREDICT (HDP of PCA REsolution and Discovery for Immunosignature Classification Testing) and HB-PREDICT (HDP of Beta PDF REsolution and Discovery for Immunosignature Classification Testing) algorithms. In a clinical setting a medical professional is only interested in whether or not a patient has a particular disease, hence the desire for only two clustering results (the patient has the disease or the patient does not have the disease). However, in the case where disease relationships need to be understood, several sub-groups or additional clusters may be useful. These could correspond to further biological states such as having previously had the disease, or perhaps instances where the patient is somewhere in the disease response continuum, as in the are of time course data.

Given these assumptions and limitations to the model, further refinement is possible. The inclusion of the weight matrix along with some additional biologic information on the binding effects of the antibodies themselves could lead to a more refined model, especially where further analysis of the peptides themselves is desired. In the case of patient classification, additional clusters could be identified and tied to other biological states in patient pathology, for example those corresponding to time effects in disease states. Given the basis for these models, further discussion of each method will be undertaken next.

7.2 Single State Comparison to Naive Bayes Classification

For comparison purposes in the area of single disease results, the same single data sets were analyzed using the naive Bayes approach indicated in [7]. This method was selected for comparison because it represented the highest performing method indicated amongst the presented algorithms. Peptides were selected using the ANOVA method and at the ~ 200 peptide significance level, though due to p-value thresholding dataset 1 only had 198 peptides and dataset 2 only had 197 peptides. Two methods were selected to create training sets for this method. Due to limited data availability for some of the diseases, the first training set was constructed as a circular moving average [115] (CMA) of four of the five immunosignaturing microarrays for each disease. This can also be thought of as leave one out averaging where for each entry in the training set for each disease state, the average of all but one (the one currently being classified) of the microarray median intensity data is used. This creates a conservative comparison by which to compare the other single disease state immunosignaturing results, as not much variation exists between the training set and actual datasets for the naive Bayes implementation which theoretically will provide good classification results. The results for each dataset are given in Figure 7.1. The resulting confusion matrix for each dataset is given in Tables 7.1 and 7.2. The second training set was contrived by using leave one out cross validation (LOOCV) [7, 116–118]. These results are also given in Figure 7.1, and the corresponding confusion matrices are given in Tables 7.3 and 7.4. Note that for these cases the SIC and RIC criteria are differently described. In this case, CMA is considered SIC while the RIC is considered the LOOCV case. Dataset 1 had a correct classification rate of 80% and 40% while dataset 2 had a correct classification rate of 84% and 84% for the CMA and LOOCV methods respectively. The better of the two classification results for each dataset will be used for comparison purposes.



(a) Dataset 1 with CMA.

(b) Dataset 2 with CMA.

(c) Dataset 1 with LOOCV.

(d) Dataset 2 with LOOCV.

Figure 7.1: Naive Bayes classification of single disease datasets.

Class	C_1	C_2	C_3	C_4	C_5	C_6
C_1	**5/5**	0/5	0/5	0/5	0/5	0/5
C_2	0/5	**4/5**	0/5	1/5	0/5	0/5
C_3	1/5	0/5	**3/5**	1/5	0/5	0/5
C_4	0/5	0/5	0/5	**4/5**	0/5	1/5
C_5	0/5	0/5	0/5	0/5	**3/5**	2/5
C_6	0/5	0/5	0/5	0/5	0/5	**5/5**

Table 7.1: Confusion matrix for Dataset 1 using naive Bayes with CMA.

Class	C_1	C_2	C_3	C_4	C_5
C_1	**5/5**	0/5	0/5	0/5	0/5
C_2	0/5	**2/5**	0/5	0/5	3/5
C_3	0/5	0/5	**5/5**	0/5	0/5
C_4	0/5	0/5	0/5	**4/5**	1/5
C_5	0/5	0/5	0/5	0/5	**5/5**

Table 7.2: Confusion matrix for Dataset 2 using naive Bayes and CMA.

Class	C_1	C_2	C_3	C_4	C_5	C_6
C_1	**3/5**	0/5	2/5	0/5	0/5	0/5
C_2	1/5	**1/5**	1/5	2/5	0/5	0/5
C_3	2/5	0/5	**2/5**	1/5	0/5	0/5
C_4	0/5	2/5	0/5	**2/5**	1/5	0/5
C_5	0/5	0/5	1/5	1/5	**2/5**	1/5
C_6	0/5	0/5	0/5	0/5	3/5	**2/5**

Table 7.3: Confusion matrix for Dataset 1 using naive Bayes and LOOCV.

Class	C_1	C_2	C_3	C_4	C_5
C_1	**4/5**	1/5	0/5	0/5	0/5
C_2	1/5	**4/5**	0/5	0/5	0/5
C_3	0/5	1/5	**4/5**	0/5	0/5
C_4	0/5	1/5	0/5	**4/5**	0/5
C_5	0/5	0/5	0/5	0/5	**5/5**

Table 7.4: Confusion matrix for Dataset 2 using naive Bayes and LOOCV.

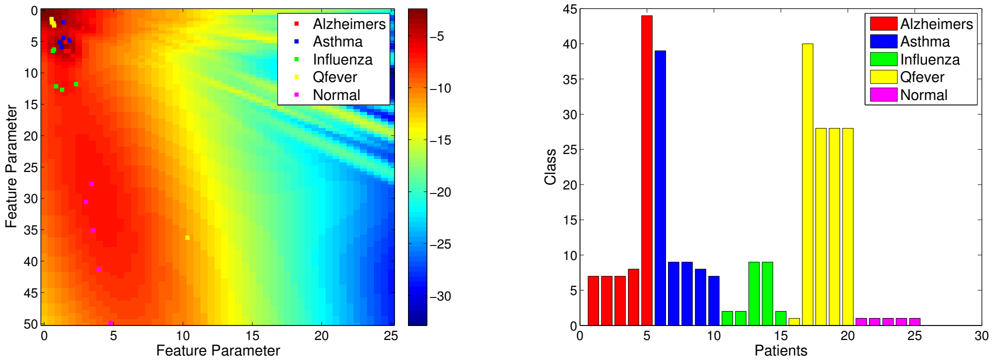
7.3 Algorithm Robustness and Sensitivity

There are several steps in the algorithm that require user input for successful execution. The first user interaction happens with the PCA visible feature and dimensionality reduction step. In this step, eigenvalue truncation happens at some user specified point thereby creating the threshold for significance. If the truncation limit is to too low, not enough data will be selected for good feature resolution. If the truncation limit is too high, little performance gain may be achieved while drastically increasing the amount of data required to analyze. As such, data is truncated at greater than 90% to help capture the majority of underlying variance in the datasets. The user must exercise some discretion in selecting the truncation limit, and some iteration may be necessary to show that sufficient information has been included. An alternative approach would be to include a performance metric that is automatically calculated and then approach the truncation limit selection like an optimization problem. However, this expansion is outside the scope of this work.

The next place user interaction may be required is in the execution of the DP-GMM. The user must specify either the total number of clusters (M) or must specify an error threshold, in addition to the innovation parameter α and base parameters. These selections drive the clustering behavior of the algorithm. If one selects values that are very restrictive, it is possible to achieve a new cluster for every single data point, while values that are not sufficiently restrictive may lead to only a single cluster being identified. While this may seem like a highly variable process, some solutions will better describe the data.

This analysis is performed by simply inspecting the data and clustering comparison. Note that this comparison does not require that one know the true underlying state in order to determine if the clustering solution is a good match. This simply requires inspecting the created clusters and determining if the clusters accurately reflect the feature space behavior. An example of the clustering results overlaid in the feature space may be seen in Figure 7.2. As one can see from this image, the estimated Gaussian distributions roughly match with the data. When considering the clustering of the latent BPFA feature space, this is demonstrated in the heuristic clustering methods presented in the Z-PREDICT and ZB-PREDICT algorithms. While it may be argued that such inspection violates the theoretical purposes behind using these adaptive methods, the practical application of such algorithms does not happen in a vacuum. Data need not be inspected or analyzed by human intervention, but could be done via some broader algorithm analysis. For example, one could compute the number of points in each cluster and determine the naturing of the clustering result.

Another place in which the user interaction will be required is the selection of parameters for the BPFA. Similar to the DP-GMM, parameters may be set so restrictively that all features are shared or that no features are shared. However, with the BPFA it was observed that when moving away from optimal solutions the



(a) BPF features on DP-GMM results.

(b) Classification results.

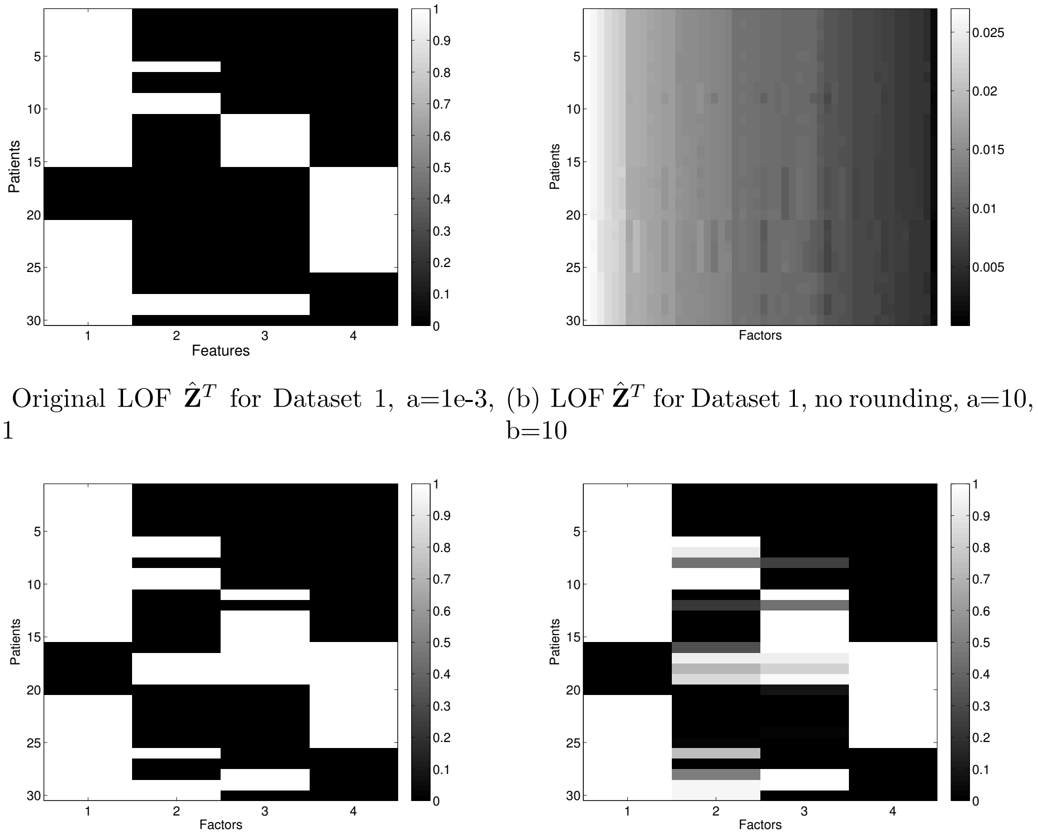
Figure 7.2: B-PREDICT results for Dataset 2.

Z matrix becomes highly variable. This can result in all features being zero or one regardless of the K truncation value chosen. This simply means that the parameters selected do not allow for good estimation given the dataset. A secondary metric that may be used to determine if adequate parameters have been selected is to look at the ordered values of π values and determine if only a few significant values are present while the others are close to zero. However, once the user is in the approximate vicinity of optimal parameters some variation can be seen in the Z matrix results. In this case, one cannot directly inspect the feature space to determine if parameter selection is adequate due to the fact that this is a latent feature space that is not observable to the user. As such, one must pay careful attention to the Z matrix. In the case of the data examined here, very little variation was seen in features, with only a few appearing or disappearing when the initial parameters were changed. However, this does not mean that this will be the case for all data. An example of feature space comparisons is given in Figure 7.3. Note that when the features vary by an order of magnitude, the determined feature space does not drastically change. This can also be seen by examining the unrounded results for the same parameters. However, when the parameters are very drastically changed, the ability to resolve individual features is lost. A large number of small valued features may be seen throughout the $\hat{\mathbf{Z}}$ matrix. Note that other than the a and b parameters, all other features were held constant: $K = 50$, and the number of burn-in and sample iterations were 2000. One way to overcome the issue of not being able to observe the latent feature space directly would be to include data whose ground state is known, such as in the multi-disease state option, and optimize for the known data. This provides the opportunity to narrow the feature space while simultaneously examining the performance. This may also be done automatically through an optimization problem where known data performance is monitored and set up as an optimization problem. This is outside the scope of this work as it was not immediately necessary or heuristically performed by the user, such as in the case of Z-PREDICTn and ZB-PREDICTn.

Finally, it should be noted that since most of the model estimation for all algorithms are MCMC based algorithms, slightly different results may be obtained by running the algorithms multiple times with exactly the same parameters. This is due to the fact that different chains will iterate towards slightly different estimations. One way to overcome this issue is to run the same parameters with multiple chains and then analyze each of the results using some success criteria, such as fit to the feature space. This is why for all simulations performed in this dissertation, each MCMC step was run a minimum of three times and then the best solution was selected.

7.4 Multi-State Comparison to the Single State Limits

In order to understand the multi-state data combined performance results, it is helpful to compare other algorithms of similar functionality. However, for immunosignaturing data of this type, no prior work was available for multi-state analysis. As such, we consider the improvement over the theoretical best performance that can be achieved in the case of single state analysis. However, if the multi-state algorithms



(a) Original LOF $\hat{\mathbf{Z}}^T$ for Dataset 1, a=1e-3, b=1

(b) LOF $\hat{\mathbf{Z}}^T$ for Dataset 1, no rounding, a=10, b=10

(c) LOF $\hat{\mathbf{Z}}^T$ for Dataset 1 a=1e-4, b=1

(d) LOF $\hat{\mathbf{Z}}^T$ for Dataset 1, no rounding, a=1e-4, b=1

Figure 7.3: ZB-PREDICT binary matrix results for various parameters

perform worse than if a single state version were used in its place, no improvement can be claimed. We define two terms to describe the theoretical best performance limit for multi-state analysis when analyzed with single-state methods.

The first term is the "Single State Upper Bound," or SSUB, which is simply the highest possible true positive and true negative results that could be achieved for n underlying states when analyzed with a single state method, since only one state may be identified in the single state methods. As such, the "Single State Lower Bound," or SSLB, is then defined as the lowest type I and type II error that could be achieved given the SSUB. The SSUB and SSLB refer to the combined true positive/true negative or combined type I/type II error percentages respectively. To find these values, it is first necessary to examine the true positive, true negative, type I and type II error values, as can be seen in the first four columns of Table 7.5, representing the limits when this is performed over Ψ key comparisons. This can give more information than the combined SSLB and SSUB values in that certain algorithms may be more adept at achieving good results in one of the categories at the expense of the others. For example, an algorithm may achieve good true positive or negative results but at the expense of some of the error terms. The combined correct results, or SSUB, and the combined error, or SSLB, over all comparisons are then given in the last two columns. Note that this arrangement is true under the assumption that all N datasets have n underlying states. It should also be noted that, unlike previously discussed multi-state algorithms, the ”no result” category is not factored in due to the underlying assumption of perfect clustering, but only single state identification. In reality, perfect clustering is not likely given the methods discussed previously.

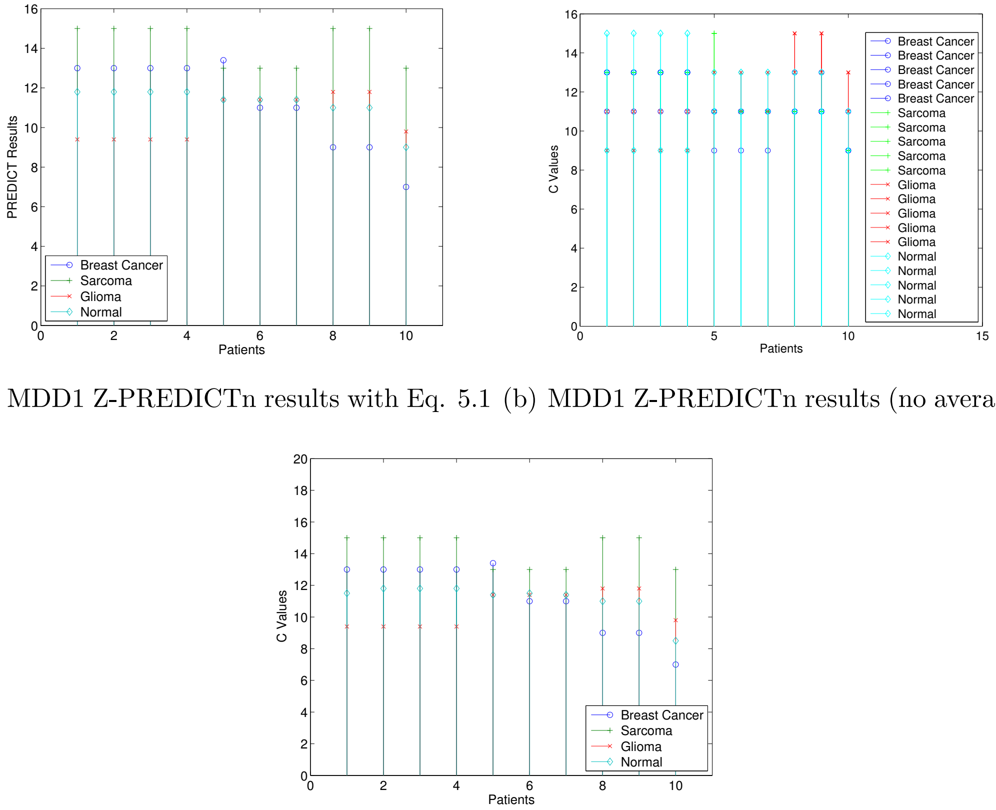
True +	**True -**	**Type I**	**Type II**	**SSUB**	**SSLB**
$\frac{N}{nN}$	$\frac{N\Psi - nN}{N\Psi - nN}$	$\frac{0}{n\Psi - nN}$	$\frac{nN - N}{nN}$	$\frac{1+\Psi - n}{\Psi}$	$\frac{n-1}{\Psi}$

Table 7.5: Single state limits

7.5 Multi-Disease Dataset Training Set Comparison

While the single state clustering methods do not require a comparison dataset to perform the clustering, training datasets would be required in order to complete the classification. While this would be relatively straightforward by including immunosignature data for known diseases and running it in parallel through the various single state algorithms, it would still create a semi-supervised environment. Similarly, the multi-disease methods require known immunosignature data in order to complete the clustering. However, in this case the data is an integral part of the algorithm, though it is not necessarily used in the adaptive portions for model estimation. As such, some discussion is useful in terms of the performance seen in previous results in order to understand the impact to the method.

Consider MDD1; previous results have a low number of available datasets for several of the diseases, and thus removing any one immunosignature datapoint may have critical impact on the performance. For instance, the results for the **C** matrix are given in Figure 7.4(a). There is some redundancy in the data points due to the low number of available datasets (the sarcoma and a single entry in the glioma key datasets). This represents a full averaging of all available datasets in the key comparison data. In order to see this impact, the averaging step, Equation (5.1c), is removed from the Z-PREDICTn algorithm and all datasets are compared to every key. This result in shown in Figure 7.4(b). While this provides a more complete picture of the interactions, the thresholding becomes more difficult in that many of the data points overlap, where as previously only the top results in each case need to be considered. In this figure, there are quite a few top results present. As such, the top two values for each patient were considered. Given this complexity, we want to unite the two approaches so that analysis is simplified. As such, the CMA approach discussed in section 7.2 is used here these results are represented in Figure 7.4(c). There is not much difference between this and Figure 7.4(a), meaning the full averaging approach is nearly equivalent to the CMA approach. The results for these three training set methods are presented in Table 7.6. For the full averaging training data, the combined correct classification was 95% with a total combined error of 5%. For the training data with no averaging, the combined correct classification was 68% with a total combined error of 32%. For the CMA training data the combined correct classification was 92.5% with a total combined error of 7.5%.



(a) MDD1 Z-PREDICTn results with Eq. 5.1 (b) MDD1 Z-PREDICTn results (no average)

(c) MDD1 Z-PREDICTn results using CMA

Figure 7.4: **C** results comparisons for various training sets

Disease	**True +**	**True -**	**Type I**	**Type II**
Full Averaging	20/20	18/20	2/20	0/20
No Averaging	76/100	60/100	40/100	24/100
CMA	19/20	18/20	2/20	1/20

Table 7.6: Indication and error rates for dataset 1 after Z-PREDICTn

7.6 Results summary table

Tables providing the classification results for all methods are given in Tables 7.7 and 7.8. Note that combined correct classification and combined error rates are given for the multi-disease datasets.

Method	**Dataset 1**		**Dataset 2**	
	SIC	**RIC**	**SIC**	**RIC**
PREDICT	60%	60%	64%	64%
B-PREDICT	70%	76.7%	64%	64%
Z-PREDICT	60%	73.3%	88%	88%
ZB-PREDICT	83.3%	83.3%	76%	76%
Φ-PREDICT	43.3%	53.3%	60%	68%
BIO-PREDICT	66.7%	66.7%	88%	88%
BIOB-PREDICT	83.3%	83.3%	76%	76%
Naive Bayes	80%	40%	84%	84%

Table 7.7: Single state algorithm clustering performance comparison.

Method	**MDD1**		**MDD2**	
	Correct	**Error**	**Correct**	**Error**
Z-PREDICTn	95%	5%	87.5%	12.5%
ZB-PREDICTn	65%	35%	100%	0%
H-PREDICT	75%	12.5%	57.5%	20%
HB-PREDICT	50%	25%	70%	15%
Single State Limits	75%	25%	75%	25%

Table 7.8: Multi-state algorithm classification performance comparison.

Method	**Breast Cancer**	**Normal**	**Glioma**	**Cocci**	**Sarcoma**	**Asthma Post**
PREDICT SIC	4	0	4	5	5	0
PREDICT RIC	4	0	4	5	5	0
B-PREDICT SIC	5	2	4	3	5	2
B-PREDICT RIC	5	2	4	5	5	2
Z-PREDICT SIC	3	3	3	2	5	2
Z-PREDICT RIC	5	3	3	4	5	2
ZB-PREDICT SIC	5	3	5	5	5	2
ZB-PREDICT RIC	5	3	5	5	5	2
PHI-PREDICT SIC	2	1	1	2	5	2
PHI-PREDICT RIC	2	3	2	2	5	2
BIO-PREDICT SIC	0	4	3	5	5	3
BIO-PREDICT RIC	0	4	3	5	5	3
BIOB-PREDICT SIC	5	3	5	5	5	2
BIOB-PREDICT RIC	5	3	5	5	5	2

Table 7.9: Results summary by disease for Dataset 1

Method	**Alzheimer's**	**Asthma**	**Influenza**	**Qfever**	**Normal**
PREDICT SIC	3	3	1	4	5
PREDICT RIC	3	3	1	4	5
B-PREDICT SIC	3	2	3	3	5
B-PREDICT RIC	3	2	3	3	5
Z-PREDICT SIC	4	4	5	4	5
Z-PREDICT RIC	4	4	5	4	5
ZB-PREDICT SIC	5	1	4	4	5
ZB-PREDICT RIC	5	1	4	4	5
PHI-PREDICT SIC	1	1	4	5	4
PHI-PREDICT RIC	1	1	5	5	5
BIO-PREDICT SIC	4	4	5	4	5
BIO-PREDICT RIC	4	4	5	4	5
BIOB-PREDICT SIC	5	1	4	4	5
BIOB-PREDICT RIC	5	1	4	4	5

Table 7.10: Results summary by disease for Dataset 2

7.7 Method results discussion

Considering Table 7.7, one can see that no method provides 100% correct classification in all datasets. As such, there are tradeoffs for each method. First considering the naive Bayes approach, this method requires a training set regardless of whether the CMA or LOOCV approaches are selected. The LOOCV approach as provided in literature was expanded to a CMA approach in order to determine if there were more optimal training sets that could provide improved performance understanding. The CMA approach performed as well or better than the LOOCV approach for both datasets. Both these methods have the benefit of providing diagnosis information using a straightforward approach with fairly low computational complexity. However, the inherent tradeoff with these approaches are that they are supervised and good for single classification assignments. If a novel pathogen response is detected, it will automatically attempt to bin it into one of the known disease states present in the training data.

While it may be possible to contrive methods that expand on the naive Bayes approach in order to compensate for this, such as thresholding the resulting comparison values in the algorithm, multiple novel biothreats would all be lumped together. Additionally, since a training set is required, the clustering is highly dependent on the use of a good training set. For some of the data analyzed, only a limited number of immunosignatures were available. Furthermore, there is concern that the training data will need to take into account not only all known diseases for classification, but also be representative of a variety of pathological responses for one particular disease state. It is conceivable that even within a patient population that the manifestations of a single disease may take on a variety of immune responses, and accurately capturing these in the training data would be required for accurate classification.

This then leads to the necessity for unsupervised methods that are capable of detecting varied immunological responses without the need for comprehensive training data. These methods are represented by the remaining methods detailed in Table 7.7. As one can see from these results, no method resulted in perfect clustering and that some results may be disease state dependent. Additionally, some methods performed better, as well as, or worse than the CMA and LOOCV naive Bayes embodiments. However, in general, as the methods grow in complexity, their clustering results improve.

When considering the PREDICT and B-PREDICT methods, one can see that there may be a slight improvement when using the B-PREDICT approach. However, for reasons explained in section 7.1, it is important to retain both methods to have a variety of approaches for the corresponding variety of biological responses that may be received. Furthermore, these two methods provide a great way to downselect the data dimensionally from $\sim$ 10000 unique sequences, to some lower number of significant data points, though it should be cautioned that the term ”significant” may also be subjective. It is also worth noting that though the results vary from 60% to ~76%, these two methods represent the lowest computational requirements for the adaptive learning algorithms presented here. Thus, it may be possible that if the immunosignaturing approach was downsized for applications where on-the-fly analysis was required, such as battlefield or rural settings where sophisticated equipment is scarce and diagnosis time is critical, this may be an acceptable first step in helping diagnose pathogens. Additionally, it may be possible to improve this method by incorporating additional patient data, such as symptoms or time of infection, to extend the multidimensional data arrays prior to input into the DP-GMM. As such, these methods may still have important applications in non-lab based settings. However, we want to improve upon these methods for better clustering.

This then leads to the Φ-PREDICT, Z-PREDICT, and ZB-PREDICT methods, whose common adaptive step is the BPFA. From the results presented in Table 7.7, Φ-PREDICT performs the worst out of these three methods. This could be due to the fact that while the latent feature determination is adaptive, the subsequent clustering analysis is not and requires additional subjective user input for both the quantization levels as well as the definitions of SIC and RIC. Because these can be quite varied, and because a fairly conservative approach was discussed in section 3.6.3, somewhat low correct clustering results are presented here. However, this does not mean that this method cannot be expanded or still useful in a lab based setting. This method provides the unique ability to not just show feature presence in a binary sense (as with $\hat{\mathbf{Z}}$ based methods), but to show feature presence linked to quantity. This can help to detect more subtle changes in the antibody profiles rather than simply a binary on/off state analysis. It should also be noted that it may be possible to improve this classification by the application of the DP-GMM rather than a user specified quantization scheme. Additionally, this method employed PCA and did not delve into a similar method that would employ the BPF dimensionality reduction scheme. These were not pursued due to the increased performance gains found when using the Z-PREDICT and ZB-PREDICT methods, which inherently required less user interaction.

In both Z-PREDICT and ZB-PREDICT, the BPFA is used to determine underlying features that are not directly discernible to the user. However, it does this by focusing on the binary presence summary indicated by the $\hat{\mathbf{Z}}$ matrix. This produces immense gains over the Φ-PREDICT method and was as good or better than either the PREDICT or B-PREDICT methods, and in some cases the naive Bayes approaches. While these are more computationally intensive, they do provide additional insight into the binding behavior for the various disease states, which may be reflected in the performance of each algorithm on the different disease datasets. This again highlights the need to have the adaptive approach be flexible enough to capture varied pathology responses while still providing good clustering. The main issue with these three methods, however, is the fact that the clustering itself is not adaptive. While this is inherently not a problem, as the datasets grow in both the number of patients and possibly even the number of peptides analyzed, there is a need to constantly adapt the output to the changing parameters. As such, adaptive clustering expansions of the Z-PREDICT and ZB-PREDICT methods were proposed.

These expansions are called the BIO-PREDICT and BIOB-PREDICT methods, and their results are comparable to those achieved in Z-PREDICT and ZB-PREDICT, as seen in Table 7.7. This helps to link together the adaptive determination of underlying or latent features with clustering based on those features. What is important to note is that this requires an expanded view of the BPFA in that single features are not indicative of single disease states. Rather, multi-feature combinations are indicative of disease states. As such, the feature spaces are treated as multidimensional spaces for input into the DP-GMM. The downside is that now two MCMC techniques are required, and the computational complexity of execution increases.

It is useful to note that the maximum amount of clusters that may be represented by K features from the BPFA is 2^K, meaning that even if too few features are selected for BPFA analysis or if the values of a and b are set to cause a high amount of feature combination, it may still be possible to obtain resolution between disease states. However, while the DP-GMM only needs to estimate a single weight for each class as defined by M, a precision, and a mean value for each data point (per Algorithm 1), the number of items to estimate for each of the patients in the BPFA is very dependent on the value selected for K. As such, DK values are estimated for the $\mathbf{\Phi}$ matrix, KN values are estimated for the $\mathbf{Z}$ matrix, and a single value for σ_n^2

for the error. Thus, as the number of features specified for the BPFA increases, the number of estimate computations greatly increases.

Furthermore, it is important to have sufficient data present such that the BPFA achieves good resolution. This can be improved and even expedited by good selection of the priors. While the same is true for the DP-GMM, the rate of estimated values increases at a slower rate defined by the truncation factor M. However, these two methods performed comparably to the naive Bayes approaches, but did not require training sets in order to perform the clustering. It is worth noting as well that there are methods in literature that exist to combine the DP-GMM and BPFA, but had this approach been used first, the view of multiple underlying features corresponding to a single disease may have been lost. While this may not have been critical when clustering the immunosignatures, it is critical if one ever has the desire to link the features back to the biological effects happening amongst the peptide groups. It is also critical if there is a desire to reward or penalize certain behavior based on the tolerance for error in the end use application. However, the combination of these two methods may be possible improvements for future work.

Each of the methods described previously (naive Bayes, PREDICT, B-PREDICT, Φ-PREDICT, Z-PREDICT, ZB-PREDICT, BIO-PREDICT, and BIOB-PREDICT), have a common underlying assumption in their execution. This assumption is that all patients have only a single disease state at a given time, which may be too restrictive in practice, especially in a clinical setting. Furthermore, with the exception of naive Bayes, these methods do not provide the critical disease diagnosis link that is desired (i.e. they do not inherently perform classification without the introduction of known disease state data). While it is possible that known immunosignatures may be included to help identify each cluster (and even possibly eliminate the need for separate SIC and RIC criteria), this was not undertaken due to the recognized restrictiveness of the single disease state requirement. However, this would not impact the detection of a novel biothreat in a population, for which immunosignatures may not be available.

In order to expand upon the single disease restriction, but to also retain the adaptive learning elements of these methods, four new method were proposed. These include Z-PREDICTn, ZB-PREDICTn, H-PREDICT, and HB-PREDICT. The results for these multi-disease embodiments maybe be seen in Table 7.8. The Z-PREDICTn and ZB-PREDICTn are direct expansions of the Z-PREDICT and ZB-PREDICT methods previously discussed, in that they now incorporate known immunosignatures for which to compare patient data. This requires further expansion of the BPFA results to now include the possibility that the feature combinations are indicative of single disease states, and also that combinations of these combinations may be indicative of multiple underlying states. This is yet another reason why the direct linkage between the BPFA and other adaptive classification methods may be difficult to implement for immunosignaturing. These types of relationships may have been missed and thus resulted in single disease classification had they been combined directly or had the Z-PREDICT or ZB-PREDICT methods been used blindly.

It should also be noted that the Z-PREDICTn and ZB-PREDICTn methods still improve upon the CMA and LOOCV naive Bayes methods in that multiple diseases may now be identified. As one can see, these methods provide fairly good classification results, but at the expense of needing additional information in order to identify present diseases, notably that one must know the number of underlying diseases. This could also be considered a thresholding problem where the user may specify a level of significance in order to identify the present diseases. It may also be possible to modify the naive Bayes approach to perform in a similar manner by avoiding the final group assignment in the algorithm, and simply plotting the comparisons made to the training data and then imposing the user specified threshold. However, this was not pursued due to the user specified threshold requirement.

Two additional methods were proposed: H-PREDICT and HB-PREDICT. These methods seek to adaptively classify the values achieved when comparing the known key immunosignatures with those of unknown patients. Due to the variation seen in the comparison values for the various patients, it is possible that the classification results will be lower than that for Z-PREDICTn and ZB-PREDICTn, which is reflected in Table 7.8. Note that it would be possible to apply the DP-GMM to the results of the Z-PREDICTn and ZB-PREDICTn algorithms, but not directly. In order for that occur, patients must not be seen as a multidimensional dataset, and all data must be combined into a single dimension disease set. While this is not impossible, it does require additional bookkeeping in order to glean patient diagnosis information. As such, the HDP is explored instead, where by the patient data is still seen as a single dimension dataset, but now is separated into patient corpora. This then eliminates the need for additional data combination and separation steps that would be required for the DP-GMM. Additionally, this helps to link formally the possible combination of the HDP and the BPFA. This also allows for the expansion of patient data sets to include other information (such as symptoms or time point data) to improve classification. The downside to the H-PREDICT and HB-PREDICT methods are that they have higher computational complexity than the Z-PREDICTn and ZB-PREDICTn methods due to the adaptive learning for diagnosis.

All methods as listed in Tables 7.7 and 7.8 result in the creation of an adaptive learning framework for immunosignaturing with improved detection and diagnosis capability. While touched upon briefly in this section, there is still much work to be done. Possible extensions will be further discussed in the next section.

Chapter 8

CONCLUSIONS AND FUTURE WORK

8.1 Conclusion

Given the methods summarized in Tables 7.7 and 7.8, we have shown that it is possible to analyze immunosignaturing data using Bayesian nonparametric adaptive learning techniques to facilitate the identification and clustering of single disease state data without the need for a training set or supervised learning techniques, as well as shown that it is possible to diagnose individuals with multiple disease pathologies. This is done while maintaining method flexibility to account for varied biological responses both on an individual level as well as within a patient population. Furthermore, the adaptive framework comprising all previously discussed methods provides algorithms with a variety of computational complexities. In Table 7.7 the methods, with the exception of the previously reported naive Bayes method, are presented in order of increasing complexity. The tradeoff presented is generally that the greater complexity yields better performing clustering. For example, PREDICT (PCA REsolution with DP-GMM for Immunosignature Classification Testing) and B-PREDICT (Beta PDF REsolution with DP-GMM for Immunosignature Classification Testing) are the methods with the lowest overhead as they contain the dimensionality reduction and Dirichlet process Gaussian mixture modeling (DP-GMM) applications only, and are simultaneously one of the lowest performing methods (60%-76.7%). The Z-PREDICT (Z-matrix from PCA REsolution and Discovery for Immunosignature Classification Testing), ZB-PREDICT (Z-matrix from Beta PDF REsolution and-Discovery for Immunosignature Classification Testing) and Φ-PREDICT (**Φ** PCA

REsolution and Discovery for Immunosignature Classification Testing) methods are the next most computationally intensive methods, but yield some of the best performance results (43.3-88%), with the lowest performing method being due to required user specified significance levels. While Z-PREDICT and ZB-PREDICT yield good results, they require some additional **Z** matrix manipulation after the fact. This can be eliminated by the application of the BIO-PREDICT (BPFA Including prOcessing with PCA REsolution and DP-GMM for Immunosignature Classification Testing) and BIOB-PREDICT (BPFA Including prOcessing with Beta PDF REsolution with DP-GMM for Immunosignature Classification Testing) methods, which automate this step. As such, the BIOB-PREDICT and BIO-PREDICT methods have some of the best clustering results (66.7%-88%), but are also the most computationally complex.

Inherent in all of these approaches is the assumption that each immunosignature should be grouped in a single cluster. This may be overly restrictive when considering that patients may have multiple pathologies, or in cases where additional data, such as disease stage or time point, is required. Furthermore, it would be helpful to combine cluster identification, such as disease diagnosis in the case of multiple diseases, with the method. This then leads to the Z-PREDICTn (Z-matrix from PCA REsolution and Discovery for Immunosignature Classification Testing up to n-states) and ZB-PREDICTn (Z-matrix from Beta PDF REsolution and Discovery for Immunosignature Classification Testing up to n-states) methods. These methods achieved fairly good results when tested on combination data (65%-100%) with fairly low error (5%-35%). While these methods do require the adaptive BPFA in execution, they do not cluster automatically. Additionally, since these methods incorporate a training set but this training set is only used in the final comparison steps of the algorithm outside of the adaptive learning framework, these are considered semi-supervised. The ability to adaptively classify based on the multi-cluster data is then explored, resulting in the H-PREDICT (HDP of PCA REsolution and Discovery for Immunosignature Classification Testing) and HB-PREDICT (HDP of Beta PDF REsolution and Discovery for Immunosignature Classification Testing) methods. While these methods resulted in slightly lower classification results due to less known user information as compared to the Z-PREDICTn and ZB-PREDICTn methods, they also extended the adaptive framework to diagnosis capabilities. This resulted in correct classification in the range of 50%-75% with combined errors in the range of 12.5%-25%. Additionally, the use of the Z-PREDICTn, ZB-PREDICTn, H-PREDICT, and HB-PREDICT methods allows for application to multiple different research problems outside of immunosignaturing as well.

The combination of these methods results in a flexible framework for adaptive clustering and ultimately diagnosis that is useful both in cases where novel data point introduction is possible with no known prior information (such as in the case of a new disease in a population), as well as in cases where cluster identification is required and multiple cluster membership is desired (such as in the case of multiple disease presence). This framework is flexible enough to incorporate additional data outside of immunosignaturing, and even be used for other unrelated problems where the researcher desires similar outcomes to those discussed here. This then represents an improved modeling and diagnostic framework.

8.2 Future Work

While this provides a foundation for the use of adaptive methods in immunosignaturing and helps to expand the area of latent feature combination clustering, especially in the case of multi-cluster membership and identification, there is still much more to be explored in these areas. First, for the single cluster membership case, the Φ-PREDICT method may be expanded to include adaptive learning clustering, thus eliminating the need for user specified thresholds and quantization. This could possibly help increase the accuracy, as well as determine an amount of membership for various features, as opposed to the binary case when only the **Z** matrix is considered.

For dimensionality reduction, other methods may be more appropriate than simply distribution fitting or PCA. Single distribution parameters may miss multimodal embodiments, and PCA supposes that all relevant data is maximized along the selected orthogonal bases. Other dimensionality reduction schemes may be more appropriate, or even combinations of these schemes, such as the use of both PCA and BPF, may result in improved performance. Further, each application involving the use of the DP-GMM, BPFA, and HDP models may select different priors that could result in further clustering or diagnosis improvements. The application of different base distributions as well as the exploration of alternate MCMC methods to improve efficiency are both possible directions. It may also be possible to set up these improvements as an optimization problem where the previously user specified parameters, such as innovation parameters, are now optimized in process, rather than remaining static throughout the estimation steps. Additionally, it may be desireable to extend the BPFA models to include feature weighting, though some additional thought is required for the biological representation behind this weigh matrix.

Additional model refinements may also be possible to better represent the biological mechanisms at play in antibody binding. For instance, this work simplifies the approach as a first step to include the assumption that only non-competitive effects are dominant, which may not be the case depending on the antibodies present or the peptide sequences used. This may be one way to incorporate a weighted model, with possible negative numbers indicating competitive effects. Additional studies may also be done to show the biological relationships between diseases, resulting in phylogenic trees for various diseases. This could be accomplished by doing a comparison between model parameters and present features for similarly related diseases and thus extended to show latent relationships between diseases previously thought unrelated. Further work may also elect to focus on specific peptides present in various disease states. In fact, these same methods may be able to be used with minimal modification, with the exception of transposing the initial input data matrix (i.e. the immunosignature data). This could help to better understand specific antibodies, and lead to refining of microarrays for specific diseases.

In the case of multi-cluster membership for a single patient, additional research is also possible. In the most straightforward case, additional data may be used (such as time course data, symptom data, etc) to show relationships between these different metrics. Interestingly, this data can be included at a variety of steps in the algorithms including at the data reduction step as simply another dimension, post-dimensionality reduction as another dimension, post BPFA as another feature (albeit not necessarily a latent one), or pre-HDP to create a multidimensional dataset as input. It may also be possible to achieve improved classification results by the adoption of a different **Z** matrix modification scheme. The method presented here is related to a non-normalized dot product, which may not be the most optimal. It is possible that one could combine this step with the naive Bayes approach but rather than assigning the datapoint to the top comparison value, simply use the entire set of comparison values as a cohort input to the HDP. The flexibility of the framework presented here allows for a variety of methods to be tried and optimized without compromising the final diagnosis intent in the algorithm. Another possible direction is the use of the hierarchical BPFA as opposed to a single BPFA dataset, which would then allow for generalization of input datasets. However, this would require an adjustment in the **Z** matrix modification that occurs pre-HDP. This could also be investigated as an optimization problem with a variety of available input datasets.

Finally, it is possible to use these methods in a variety of problems outside of immunosignaturing. These methods would be most helpful in high dimensionality data situations where there is a desire to adaptively learn, cluster, and classify the input data. This can included cases where single cluster membership is desired, as well as cases were more complex classification relationships need to be understood, such as in the case of multi-cluster assignment.

Given this discussion, there are a variety of directions that can be pursued using these methods. As such, this work is seen as a solid initiation point with further classification and computational improvements possible. This can help to further improve the immunosignaturing platform as well as extend to other cases where there is a desire to link multi-feature analysis with multi-membership classification.

REFERENCES

[1] X. Duburcq, C. Olivier, F. Malingue, R. Desmet, A. Bouzidi, F. Zhou, C. Auriault, H. Gras-Masse, and O. Melnyk, "Peptide-protein microarrays for the simultaneous detection of pathogen infections," *Bioconjugate Chemistry*, vol. 15, no. 2, pp. 307–316, 2004.

[2] J. B. Delehanty and F. S. Ligler, "A microarray immunoassay for simultaneous detection of proteins and bacteria," *Analytical Chemistry*, vol. 74, no. 21, pp. 5681–5687, 2002.

[3] U. Reineke, C. Ivascu, M. Schlief, C. Landgraf, S. Gericke, G. Zahn, H. Herzel, R. Volkmer-Engert, and J. Schneider-Mergener, "Identification of distinct antibody epitopes and mimotopes from a peptide array of 5520 randomly generated sequences," *Journal of Immunological Methods*, vol. 267, no. 1, pp. 37–51, 2002.

[4] K. W. Boltz, M. J. Gonzalez-Moa, P. Stafford, S. A. Johnston, and S. A. Svarovsky, "Peptide microarrays for carbohydrate recognition," *Analyst*, vol. 134, no. 4, pp. 650–652, 2009.

[5] J. Fu, K. Cai, S. A. Johnston, and N. W. Woodbury, "Exploring peptide space for enzyme modulators," *Journal of the American Chemical Society*, vol. 132, no. 18, pp. 6419–6424, 2010.

[6] K. Usui, K.-y. Tomizaki, and H. Mihara, "A designed peptide chip: Protein fingerprinting technology with a dry peptide array and statistical data mining," in *Peptide Microarrays*. Springer, 2009, pp. 273–284.

[7] M. Kukreja, S. Johnston, and P. Stafford, "Comparative study of classification algorithms for immunosignaturing data," *BMC Bioinformatics*, vol. 13, 2012.

[8] ——, "Immunosignaturing microarrays distinguish antibody profiles of related pancreatic diseases," *Journal of Proteomics and Bioinformatics*, vol. S6, 2012.

[9] J. B. Legutki, D. M. Magee, P. Stafford, and S. A. Johnston, "A general method for characterization of humoral immunity induced by a vaccine or infection," *Vaccine*, vol. 28, no. 28, pp. 4529–4537, 2010.

[10] L. Restrepo, P. Stafford, D. Magee, and S. Johnston, "Application of immunosignatures to the assessment of alzheimer's disease," *American Neurological Association*, pp. 286–295, 2011.

[11] B. A. Chase, S. A. Johnston, and J. B. Legutki, "Evaluation of biological sample preparation for immunosignature-based diagnostics," *Clinical and Vaccine Immunology*, pp. 352–358, 2012.

[12] P. Stafford, R. Halperin, J. Legutki, D. Magee, J. Galgiani, and S. Johnston, "Physical characterization of the immunosignaturing effect," *Molecular & Cellular Proteomics*, vol. 11, no. 4, 2012.

[13] C. Morales Betanzos, M. J. Gonzalez-Moa, K. W. Boltz, B. D. Vander Werf, S. A. Johnston, and S. A. Svarovsky, "Bacterial glycoprofiling by using random sequence peptide microarrays," *ChemBioChem*, vol. 10, no. 5, pp. 877–888, 2009.

[14] [Online]. Available: http://www.biodesign.asu.edu/research/research-centers/innovations-in-medicine

[15] E. D. Getzoff, H. M. Geysen, S. J. Rodda, H. Alexander, J. A. Tainer, and R. A. Lerner, "Mechanisms of antibody binding to a protein," *Science*, vol. 235, no. 4793, pp. 1191–1196, 1987.

[16] P. Stafford and Y. Chen, "Expression technology: A review of the performance and interpretation of expression microarrays," *IEEE Signal Processing Magazine*, pp. 18–26, 2007.

[17] A. C. Scheck and P. Stafford, "Design and use of biomarkers for the current and future clinical management of brain tumors," *Biomarkers in Medicine*, vol. 6, no. 3, pp. 293–295, 2012.

[18] P. Stafford and S. Johnston, "Microarray technology displays the complexities of the humoral immune response," *Expert review of molecular diagnostics*, vol. 11, no. 1, pp. 5–8, 2011.

[19] K. F. Sykes, J. B. Legutki, and P. Stafford, "Immunosignaturing: a critical review," *Trends in biotechnology*, 2012.

[20] A. K. Hughes, Z. Cichacz, A. Scheck, S. W. Coons, S. A. Johnston, and P. Stafford, "Immunosignaturing can detect products from molecular markers in brain cancer," *PloS one*, vol. 7, no. 7, p. e40201, 2012.

[21] A. Hirakawa, C. Hamada, and I. Yoshimura, "Sample size calculation for a regularized t-statistic in microarray experiments," *Statistics and Probability Letters*, vol. 81, pp. 870–875, July 2011.

[22] R. D. Wolfinger, G. Gibson, E. D. Wolfinger, L. Bennett, H. Hamadeh, P. Bushel, C. Afshari, and R. S. Paules, "Assessing gene significance from cdna microarray expression data via mixed models," *Journal of Computational Biology*, vol. 8, no. 6, pp. 625–637, 2001.

[23] J. R. Haan, S. Bauerschmidt, R. van Schaik, E. Piek, L. Buydens, and R. Wehrens, "Robust ANOVA for microarray data," *Chemometrics and Intelligent Laboratory Systems*, vol. 98, pp. 38–44, 2009.

[24] J. R. Brown, P. Stafford, S. Johnston, and V. Dinu, "Statistical methods for analyzing immunosignatures," *BMC Bioinformatics*, vol. 12, 2011.

[25] R. F. Halperin, P. Stafford, J. S. Emery, K. A. Navalkar, and S. A. Johnston, "Guitope: an application for mapping random-sequence peptides to protein sequences," *BMC Bioinformatics*, vol. 13, no. 1, p. 1, 2012.

[26] I. Priness, O. Maimon, and I. Ben-Gal, "Evaluation of gene-expression clustering via mutual information distance measure," *BMC Bioinformatics*, vol. 8, p. 111, 2007.

[27] S. Tavazoie, D. Hughes, M. J. Campbell, R. J. Cho, and G. M. Church, "Systematic determination of genetic network architecture," *Nature Genetics*, vol. 22, pp. 281–285, 1999.

[28] A. Brazma and J. Vilo, "Gene expression data analysis," *Federation of European Biochemical Societies Letters*, vol. 480, pp. 17–24, 2002.

[29] J. L. DeRisi, V. R. Iyer, and P. O. Brown, "Exploring the metabolic and genetic control of gene expression on a genomic scale," *Science*, vol. 278, pp. 680–686, 1997.

[30] K. Kroening, S. A. Johnston, and J. B. Legutki, "Autoreactive antibodies raised by self derived de novo peptides can identify unrelated antigens on protein microarrays. are autoantibodies really autoantibodies?" *Experimental and molecular pathology*, vol. 92, no. 3, pp. 304–311, 2012.

[31] M. Kathleen Kerr and G. A Churchill, "Statistical design and the analysis of gene expression microarray data," *Genetical research*, vol. 77, no. 02, pp. 123–128, 2001.

[32] M. Brown, W. Grundy, D. Lin, N. Cristianini, C. Sugnet, T. Furey, M. Ares, and D. Haussler, "Knowledge-based analysis of microarray gene expression data by using support vector machines," *Proceedings of the National Academy of Sciences*, vol. 97, no. 1, pp. 262–267, 2000.

[33] R. Kustra, R. Shioda, and M. Zhu, "A factor analysis model for functional genomics," *BMC Bioinformatics*, vol. 7, 2006.

[34] B. Y. Renard, M. Löwer, Y. Kühne, U. Reimer, A. Rothermel, Ö. Türeci, J. C. Castle, and U. Sahin, "rapmad: Robust analysis of peptide microarray data," *BMC bioinformatics*, vol. 12, no. 1, p. 324, 2011.

[35] Y. Blum, G. L. Mignon, S. Lagarrigue, and D. Causeur, "A factor model to analyze heterogeneity in gene expression," *BMC Bioinformatics*, vol. 11, 2010.

[36] B. Chen, M. Chen, J. P. Paisley, A. Zas, C. Woods, G. S. Ginsburg, A. Hero, J. Lucas, D. Dunson, and L. Carin, "Bayesian inference of the number of factors in gene-expression analysis: Application to human virus challenge studies," *BMC Bioinformatics*, vol. 11, 2010.

[37] Z. Qin, "Clustering microarray gene expression data using weighted chinese restaurant process," *Bioinformatics*, vol. 22, no. 16, pp. 1988–1997, 2006.

[38] M. Medvedovic and S. Sivaganesan, "Bayesian infinite mixture model based clustering of gene expression profiles," *Bioinformatics*, vol. 18, no. 9, pp. 1194–1206, 2002.

[39] E. Meeds, Z. Ghahramani, R. Neal, and S. Roweis, "Modeling dyadic data with binary latent factors," *Neural Information Processing Systems 19 (NIPS)*, pp. 977–984, 2007.

[40] D. A. Redelmeier, S. H. Tan, and G. L. Booth, "The treatment of unrelated disorders in patients with chronic medical diseases," *New England Journal of Medicine*, vol. 338, no. 21, pp. 1516–1520, 1998.

[41] M. E. Tinetti, S. T. Bogardus Jr, and J. V. Agostini, "Potential pitfalls of disease-specific guidelines for patients with multiple conditions," *New England Journal of Medicine*, vol. 351, no. 27, pp. 2870–2874, 2004.

[42] A. M. Behbehani, "The smallpox story: life and death of an old disease." *Microbiological Reviews*, vol. 47, no. 4, p. 455, 1983.

[43] A. Malin, J. Zhang, B. Chakraborty, N. Kovvali, A. Papandreou-Suppappola, S. Johnston, and P. Stafford, "Adaptive learning of immunosignaturing peptide array features for biothreat detection and classification," *Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR 2011)*, pp. 1883–1887, 2011.

[44] A. Malin, N. Kovvali, A. Papandreou-Suppappola, J. J. Zhang, S. Johnston, and P. Stafford, "Beta process based adaptive learning for immunosignature microarray feature identification," *Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR 2012)*, 2012.

[45] D. Niu, J. Dy, and Z. Ghahramani, "A nonparametric bayesian model for multiple clustering with overlapping feature views." *Journal of Machine Learning Research*, vol. 22, pp. 814–822, 2012.

[46] K. Heller and Z. Ghahramani, "A nonparametric bayesian approach to modeling overlapping clusters," *Eleventh International Conference on Artificial Intelligence and Statistics*, vol. 2, pp. 187–194, 2007.

[47] A. Malin, N. Kovvali, and A. Papandreou-Suppappola, "Adaptive unsupervised learning and clustering of immune responses," submitted to *IEEE Transactions on Biomedical Engineering*, 2013.

[48] A. Malin, N. Kovvali, A. Papandreou-Suppappola, B. O'Donnell, S. Johnston, and P. Stafford, "Adaptive learning of immunosignaturing features for multi-disease pathologies," *Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR 2013)*, 2013.

[49] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1, pp. 37–52, 1987.

[50] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[51] J. Shafer and K. Strimmer, "A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics," *Statistical Applications in Genetics and Molecular Biology*, vol. 4, 2005.

[52] I. T. Jolliffe, *Principal Component Analysis.* Springer, 2002.

[53] S. Gershman and D. Blei, "A tutorial on Bayesian nonparametric models," *Journal of Mathematical Psychology*, vol. 56, pp. 1–12, 2012.

[54] C. Romesburg, *Cluster Analysis for Researchers.* Lulu Press, 2004.

[55] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification.* John Wiley & Sons, 2012.

[56] D. J. MacKay, *Information Theory, Inference and Learning Algorithms.* Cambridge university press, 2003.

[57] R. Gnanadesikan, R. Pinkham, and L. P. Hughes, "Maximum likelihood estimation of the parameters of the beta distribution from smallest order statistics," *Technometrics*, vol. 9, no. 4, pp. 607–620, 1967.

[58] R. J. Beckman and G. L. TietJen, "Maximum likelihood estimation for the beta distribution," *Journal of Statistical Computation and Simulation*, vol. 7, no. 3-4, pp. 253–258, 1978.

[59] S. M. AbouRizk, D. W. Halpin, and J. R. Wilson, "Fitting beta distributions based on sample data," *Journal of Construction Engineering and Management*, vol. 120, no. 2, pp. 288–305, 1994.

[60] F. Cribari-Neto and K. L. Vasconcellos, "Nearly unbiased maximum likelihood estimation for the beta distribution," *Journal of Statistical Computation and Simulation*, vol. 72, no. 2, pp. 107–118, 2002.

[61] P. Paolino, "Maximum likelihood estimation of models with beta-distributed dependent variables," *Political Analysis*, vol. 9, no. 4, pp. 325–346, 2001.

[62] A. K. Gupta and S. Nadarajah, *Handbook of beta distribution and its applications.* CRC Press, 2004, vol. 175.

[63] D. Fink. (1997) A compendium of conjugate priors. [Online]. Available: http://www.people.cornell.edu/pages/df36/CONJINTRnew\%20TEX.pdf

[64] H. Raiffa and R. Schlaifer, *Applied Statistical Decision Theory.* Boston, Division of Research, Graduate School of Business Adminitration, Harvard University, 1961.

[65] P. M. Lee, *Bayesian Statistics: An Introduction.* John Wiley & Sons, 2012.

[66] K. Ni, Y. Qi, and L. Carin, "Multiaspect target detection via the infinite hidden Markov model," *Journal of the Acoustical Society of America*, vol. 121, pp. 2731–2742, 2007.

[67] Y. Qi, J. W. Paisley, and L. Carin, “Music analysis using hidden Markov mixture models,” *IEEE Transactions on Signal Processing*, vol. 55, pp. 5209–5224, 2007.

[68] D. Ting, G. Wang, M. Shapovalov, R. Mitra, M. I. Jordan, and R. L. D. Jr., “Neighbor-dependent Ramachandran probability distributions of amino acids developed from a hierarchical Dirichlet process model,” *PLoS Computational Biology*, vol. 6, 2010.

[69] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky, “A sticky HDP-HMM with application to speaker diarization,” *Annals of Applied Statistics*, vol. 5, pp. 1020–1056, 2011.

[70] M. D. Escobar and M. West, “Bayesian density estimation and inference using mixtures,” *Journal of the American Statistical Association*, vol. 90, pp. 577–588, 1995.

[71] D. Young, “An overview of mixture models,” *Statistics Surveys*, pp. 1–24, 2008.

[72] M. D. Escobar and M. West, “Bayesian density estimation and interference using mixtures,” *Journal of the American Statistical Association*, vol. 96, pp. 577–588, 1995.

[73] D. Gorur and C. Rasmussen, “Dirichlet process gaussian mixture models: Choice of base distribution,” *Journal of Computer Science and Technology*, vol. 25, pp. 615–626, 2010.

[74] J. Sethuraman, “A constructive definition of Dirichlet priors,” *Statistica Sinica*, vol. 4, pp. 639–650, 1994.

[75] M. Jordan, “Bayesian nonparametric learning: Expressive priors for intelligent systems,” *Heuristics, Probability and Causality: A Tribute to Judea Pearl*, pp. 167–186, 2010.

[76] T. S. Ferguson, “A Bayesian analysis of some nonparametric problems,” *The Annals of Statistics*, vol. 1, pp. 209–230, 1973.

[77] D. Blackwell and J. B. MacQueen, “Ferguson distributions via Polyá urn schemes,” *The Annals of Statistics*, vol. 1, pp. 353–355, 1973.

[78] C. E. Antoniak, “Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems,” *The Annals of Statistics*, vol. 2, pp. 1152–1174, 1974.

[79] M. West, P. Muller, and M. D. Escobar, “Hierarchical priors and mixture models, with applications in regression and density estimation,” in *Aspects of Uncertainty*, P. R. Freeman and A. F. Smith, Eds. John Wiley, 1994, pp. 363–386.

[80] H. Ishwaran and L. F. James, “Gibbs sampling methods for stick-breaking priors,” *Journal of the American Statistical Association*, vol. 96, pp. 161–173, 2001.

[81] D. Chakraborty, N. Kovvali, J. Zhang, A. Papandreou-Suppappola, and A. Chattopadhyay, "Adaptive learning for damage classification in structural health monitoring," *43rd Asilomar Conference on Signals, Systems and Computers*, pp. 1678–1682, 2009.

[82] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, *Markov chain Monte Carlo in practice.* CRC press, 1996, vol. 2.

[83] A. S. Monto, S. Gravenstein, M. Elliott, M. Colopy, and J. Schweinle, "Clinical signs and symptoms predicting influenza infection," *Archives of Internal Medicine*, vol. 160, no. 21, p. 3243, 2000.

[84] M. Jordan, "Hierarchical models, nested models and completely random measures," *Frontiers of Statistical Decision Making and Bayesian Analysis: in Honor of James O. Berger. New York: Springer*, 2010.

[85] N. Hjort, "Nonparametric bayes estimators based on beta processes in models for life history," *The Annals of Statistics*, vol. 18, pp. 1259–1294, 1990.

[86] J. Paisley and L. Carin, "Nonparametric factor analysis with beta process priors," *Proceedings of the 26th International Conference on Machine Learning*, pp. 777–784, 2009.

[87] E. Fox, E. Sudderth, M. Jordan, and A. Willsky, "Joint modeling of multiple related time series via the beta process," arXiv:1111.4226, 2011.

[88] ——, "Nonparametric Bayesian learning of switching linear dynamical systems," *Advances in Neural Information Processing Systems*, vol. 21, pp. 457–464, 2008.

[89] L. Ren, Y. Wang, D. Dunson, and L. Carin. (2011) The kernel beta process. [Online]. Available: http://people.ee.duke.edu/~lcarin/KBP_NIPS2011_Final.pdf

[90] J. Paisley, A. Zaas, C. Woods, G. Ginsburg, and L. Carin, "A stick breaking construction for the beta process," *Proceedings of the 27th International Conference on Machine Learning*, 2010.

[91] T. Broderick, M. Jordan, and J. Pitman, "Beta processes, stick-breaking and power laws," *Bayesian analysis*, vol. 7, no. 2, pp. 439–476, 2012.

[92] Z. Ghahramani, T. L. Griffiths, and P. Sollich, "Bayesian nonparametric latent feature models," in *Valencia World Meeting on Bayesian Statistics*, Benidorm, Spain, 2006.

[93] G. Polatkan, M. Zhou, L. Carin, D. Blei, and I. Daubechies, "A Bayesian nonparametric approach to image super-resolution," *arXiv preprint arXiv:1209.5019*, 2012.

[94] F. Wood, T. Griffiths, and Z. Ghahramani, "A non-parametric Bayesian method for inferring hidden causes," *arXiv preprint arXiv:1206.6865*, 2012.

[95] D. Andrzejewski, "Accelerated gibbs sampling for infinite sparse factor analysis," Lawrence Livermore National Laboratory (LLNL), Livermore, CA, Tech. Rep., 2011.

[96] R. Thibaux and M. I. Jordan, "Hierarchical beta processes and the indian buffet process," in *International Conference on Artificial Intelligence and Statistics*, vol. 11, 2007, pp. 564–571.

[97] T. Griffiths and Z. Ghahramani, "Infinite latent feature models and the indian buffet process," 2005. [Online]. Available: http://eprints.pascal-network.org/archive/00001359/

[98] M. Zhou, H. Chen, J. Paisley, L. Ren, L. Li, Z. Xing, D. Dunson, G. Sapiro, and L. Carin, "Nonparametric Bayesian dictionary learning for analysis of noisy and incomplete images," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 130–144, 2012.

[99] M. Chen, J. Silva, J. Paisley, C. Wang, D. Dunson, and L. Carin, "Compressive sensing on manifolds using a nonparametric mixture of factor analyzers: Algorithm and performance bounds," *IEEE Transactions on Signal Processing*, vol. 58, pp. 6140–6155, 2010.

[100] D. J. Sheskin, *Handbook of parametric and nonparametric statistical procedures.* crc Press, 2003.

[101] K. Okazaki and S. Takada, "Dynamic energy landscape view of coupled binding and protein conformational change: Induced-fit versus population-shift mechanisms," *Proceedings of the National Academy of Sciences*, vol. 105, no. 32, pp. 11 182–11 187, 2008.

[102] J. R. Crowther, *The ELISA guidebook.* Springer, 2000, vol. 149.

[103] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Sharing clusters among related groups: Hierarchical Dirichlet processes," in *Advances in Neural Information Processing Systems.* MIT Press, 2005, pp. 1385–1392.

[104] G. Heinrich, "Infinite LDA implementing the HDP with minimum code complexity," Technical report, arbylon.net, Tech. Rep., 2011. [Online]. Available: http://arbylon.net/publications/ilda.pdf

[105] M. Hoffman, D. Blei, and P. Cook, "Content-based musical similarity computation using the hierarchical Dirichlet process," in *Proceedings of the International Symposium on Music Information Retrieval*, 2008, pp. 349–354.

[106] E. B. Fox, E. B. Sudderth, and A. S. Willsky, "Hierarchical Dirichlet processes for tracking maneuvering targets," in *2007 10th International Conference on Information Fusion.* IEEE, 2007, pp. 1–8.

[107] K. Sirts and T. Alumäe, "A hierarchical Dirichlet process model for joint part-of-speech and morphology induction," in *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* Association for Computational Linguistics, 2012, pp. 407–416.

[108] E. Zavitsanos, G. Paliouras, and G. A. Vouros, "Non-parametric estimation of topic hierarchies from texts with hierarchical Dirichlet processes," *The Journal of Machine Learning Research*, vol. 12, pp. 2749–2775, 2011.

[109] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical dirichlet processes," *Journal of the American Statistical Association*, vol. 101, 2004.

[110] E. Xing, K. Sohn, M. Jordan, and Y. Teh, "Bayesian multi-population haplotype inference via a hierarchical Dirichlet process mixture," in *Proceedings of the 23rd international conference on Machine learning.* ACM, 2006, pp. 1049–1056.

[111] K. Sohn and E. Xing, "A hierarchical Dirichlet process mixture model for haplotype reconstruction from multi-population data," *The Annals of Applied Statistics*, vol. 3, no. 2, pp. 791–821, 2009.

[112] D. Wulsin, S. Jensen, and B. Litt, "A hierarchical dirichlet process model with multiple levels of clustering for human eeg seizure modeling," *arXiv preprint arXiv:1206.4616*, 2012.

[113] X. Wang, W. Grimson, and C. Westin, "Tractography segmentation using a hierarchical Dirichlet processes mixture model," *NeuroImage*, vol. 54, no. 1, pp. 290–302, 2011.

[114] C. Wang, J. Paisley, and D. Blei, "Online variational inference for the hierarchical Dirichlet process," in *Artificial Intelligence and Statistics*, 2011.

[115] A. W. Marshall, I. Olkin, and B. C. Arnold, *Inequalities: Theory of majorization and its applications.* Springer, 2011.

[116] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 36, pp. 111–147, 1974.

[117] ——, "Corrigenda: Cross-validatory choice and assessment of statistical predictions," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 38, p. 102, 1976.

[118] S. Arlot and A. Celisse, "A survey of cross-validation procedures for model selection," *Statistics Surveys*, vol. 4, pp. 40–79, 2010.