UNSUPERVISED DISCOVERY OF HIDDEN GEOMETRY IN HIGH DIMENSIONAL DATASETS FOR NEURONAL ARBOR ANALYTICS

A Dissertation

Presented to

The Faculty of the Department of Electrical Engineering

University of Houston

In Partial Fulfillment

Of the Requirements for the Degree of

Doctor of Philosophy

in Electrical Engineering

By

Yanbin Lu

May 2016

UNSUPERVISED DISCOVERY OF HIDDEN GEOMETRY IN HIGH DIMENSIONAL DATASETS FOR NEURONAL ARBOR ANALYTICS

Yanbin Lu

Approved:

Chair of the Committee,
Badrinath Roysam, Department Chair,
Department of Electrical and
Computer Engineering.

Committee Members:

Zhu Han, Professor,
Department of Electrical and
Computer Engineering.

Saurabh Prasad, Assistant Professor,
Department of Electrical and
Computer Engineering.

David Mayerich, Assistant Professor,
Department of Electrical and
Computer Engineering.

Navin Varadarajan, Assistant Professor,
Department of Chemical and
Biomolecular Engineering.

Jason Eriksen, Associate Professor,
Department of Pharmacological and
Pharmaceutical Sciences.

Suresh K. Khator, Associate Dean,
Cullen College of Engineering.

Badrinath Roysam, Department Chair,
Department in Electrical Engineering.

Acknowledgements

I would like to express my sincere gratitude to my advisor Dr. Badrinath Roysam for support of my Ph.D. study and research, for his patience, motivation, and immense knowledge. Besides, I would also like to thank the rest of my dissertation committee for their insightful comments and encouragement.

My sincere thanks also goes to John Ellenberger, who provided me an opportunity to join their team as an intern, and who gave the access to the resources and research facilities in SAP.

I would like to say thank you to everyone in the Bio-image Analytics Lab. We have been discussing problems for so many times, helping each other with difficulties in and out of the lab. I am fortunate to have met them and I would cherish their friendship for the rest of my life. I would like to thank all my friends, who have brought happiness and thrilling experiences to my life, and make me feel like being at home when I’m here in a foreign country.

Last but not the least, I would like to thank my parents and my sister for their support during my Ph.D. study. I thank them for the unconditional love and care at all time, being with me for all ups and downs in my life so far.

An Abstract
of a
Dissertation

Presented to

The Faculty of the Department of Electrical Engineering

University of Houston

In Partial Fulfillment

Of the Requirements for the Degree of

Doctor of Philosophy

in Electrical Engineering

By

Yanbin Lu

May 2016

Abstract

Brain cells such as microglial, neurons, and astrocytes, undergo varieties of morphological and functional changes during their life cycle, or responding to micro environmental perturbation. To better understand the functions of these cells in different status and sub-status, it is essential to study the morphological characteristics of the cells. More importantly, the process of categorizing cells into different status and sub-status serves as a prerequisite for further analysis. In this dissertation, we propose a method for comparative profiling of quantitative arbor morphology data across multiple ensembles of brain cells (neurons, glia) with the overall goal of analyzing alterations/differences in cellular arbors, in a manner that also facilitates qualitative interpretation and model inference. Our method works in three steps. First, we propose a robust unsupervised co-clustering method for the purpose of robustly identifying groups and sub-groups of cells with similar morphologies, and simultaneously identifying the hierarchical grouping patterns among the corresponding arbor measurements. Second, we compare the identified cell groups and morphological measurements using an adaptation of Tibshirani's Sparse Group LASSO algorithm, allowing us to identify the most significant feature groups, in addition to individual features that underlie the cellular arbor alterations. Finally, for the special common case when batches of images are continuously being acquired under a common imaging protocol, we describe an approach in which the first step can be performed only once, and the resulting models are reused for subsequent imaging experiments, thereby saving computing time. We illustrate our comparative arbor analytics method with data from a neuro-engineering study aimed at profiling glial alterations resulting from the insertion of a neural recording device into the brain tissue. The experiments on microglial and astrocytes perturbation analysis in the Binge Alcohol study also proofed the power of the proposed method in discovering cell types/subtypes, and discovering features of significance in real biological applications.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

The arbors of brain cells, especially neurons, astrocytes, and microglia, can undergo alterations in response to external perturbations (e.g., injury, healing, alcohol abuse, drugs), and/or exhibit differences across brain tissue samples that have been subjected to different experimental conditions (Zanier et al., 2015). There is a widespread need to detect and quantify these arbor alterations, since they are informative of the underlying biological processes (Morrison et al., 2013). Such analysis must be sensitive to subtle changes, and be scalable to large ensembles of arbors at the same time. As noted widely, there is no single measurement that can quantify an arbor, but a sufficiently comprehensive and multifaceted collection of arbor measurements can capture an arbor's morphology in the aggregate (Scorcioni et al., 2008; Parekh, et al. 2013; Lu et al., 2015). Given the sheer diversity of arbor alterations, it is also important to have the ability to separate the more significant/important changes from others. Importantly, the analysis should provide an output that allows an investigator to interpret the results at a qualitative level.

The availability of quantitative arbor measurements for large cell populations is growing rapidly, as exemplified by the NeuroMorpho database containing contributions from investigators around the world (www.neuromorpho.org). This is also true at the level of an individual experimental study. For example, advances in fluorescence confocal microscopy and stage robotics now enable large multi-millimeter swaths of brain tissue to be imaged in a step-and-repeat manner, and the resulting images can be stitched together to create seamless mosaics (montages) that capture thousands of cells at fine resolution (250 – 300nm/pixel), while preserving their spatial distributions over extended distances (Tsai et al., 2011; Rey et al., 2014). At the same time, advances in methods for large-scale automated reconstruction of cellular arbors (Megjhani et al., 2015, Kulkarni et al., 2015, Zhou et al., 2016, Peng et al., 2015, Meijering 2010) enable quantification of cellular arbors at the population scale, using comprehensive libraries of arbor features that capture detailed measurements of arbor anatomy, especially Ascoli's L-measure (Scorcioni et al., 2008, Parekh, et al., 2013) that can be exported to the open SWC data representation (Cannon et al., 1998). Finally, software tools for visualizing and editing arbor reconstructions have progressed to the point when large ensembles of reconstructions (tens of thousands), rather than just one reconstruction, can be visualized three-dimensionally, edited, and quantitatively analyzed in a collective manner (Luisi et al., 2011, www.farsight-toolkit.org). These studies produce high-dimensional datasets (~ 130 dimensions per cell) that require effective multivariate bio-informatics "big data analytics" algorithms. We have earlier called this capability "arbor analytics" (Lu et al., 2015). Recent advances (e.g., Lu et al., 2015, Xu et al., 2016) in arbor analytics make it possible to profile arbors in a comprehensive manner, and at the scale of large cell populations (e.g., thousands of cells). In this paper, we extend arbor analytics to the next logical step – comparative population-scale profiling of two or more ensembles of cellular arbors with the goal of analyzing cellular differences and alterations. For these reasons, we refer to our proposed method as "comparative arbor analytics."

In this paper, we propose a comprehensive and usable method for population-scale comparative arbor analytics that includes three major components. First, we focus on the problem of unsupervised/exploratory analysis of quantitative arbor morphology data for large cell populations (>50 cells). Specifically, we describe a method for unsupervised co-clustering of quantitative arbor morphology data (specifically, Scorcioni et al.'s L-measure) for a population of cells. This entails identifying groups of cells with similar morphologies in a hierarchical manner, and simultaneously identifying the hierarchical grouping patterns among the corresponding arbor measurements for the purpose of interpreting the characteristics of the identified cell groups. In choosing approaches to exploratory co-clustering, we have emphasized practical usability. One practical barrier to usability relates to the setting and adjustment (tuning) of parameters. These internal settings of analytics algorithms are often not understandable to non-mathematical users. In this regard, our method is capable of generating usable results without requiring any parameter settings from users. Two adjustments are optionally available to the user, and they are both quite intuitive, as described later. Another barrier to usability of cluster analysis systems is their sensitivity to noise, outliers, and data variability. Our method is designed with built-in wavelet-based "smoothing" mechanisms, and specific choices of distance metrics that allow it to detect grouping patterns in a robust manner. Finally, our method is designed to be hierarchical, so groups and sub-groups in the data can be detected effectively and rapidly.

Even after the grand data re-organization provided by the unsupervised harmonic co-clustering step, qualitative interpretation is still somewhat challenging due to the fact that the arbor measurements are a complex combination of heterogeneous quantities. For example, the L-measure data include a mix of spatial lengths, widths, and volumes, and several dimensionless quantities (e.g., counts of objects, shape features, etc.). These measurements are extracted at different scales ranging from spines of neurons to gross measurements of an entire arbor. The scientific significance of these features must be established in an intelligent and guided manner. For example, it cannot be based purely on statistical measures on individual features alone, they should be based on a thoughtful combination of statistical measures and qualitative considerations emanating from the scientific context and the objectives of the investigator. The sparse group least absolute shrinkage and selection operator (LASSO) method pioneered by Tibshirani (1997) provides us with the ability to address these compelling needs in a powerful yet practical manner. This method forms the core of the second step of our method, and it enables us to reveal a sparse set of the most important explanatory factors underlying the population-level arbor differences at chosen scales and/or levels of detail. For example, it allows us to perform a thoughtful grouping of L-measure features in a manner that facilitates human interpretation of the machine-computed results, and the sparse group LASSO enables us to identify the most distinguishing feature groups, and individual features within. The fundamental power of this method emanates from its use of the ℓ_0 norm in the data regularization. In this method, variable selection typically amounts to the selection of the important factors (groups of variables) rather than individually derived variables, as each factor corresponds to one measured variable. The resulting interpretation model allows us to perform a comprehensive comparative analysis between cell groups. It provides the ability to discover insights within each cell group as of what features are significantly different between groups.

Finally, for the practical case when batches (or a series) of images are being acquired under a common imaging protocol, typically as part of a larger ongoing study, it may be impractical to perform the above-described two-step analysis after all of the data is fully collected. Since the images are being acquired under the same conditions, it is preferable to perform the first harmonic co-clustering step once, and then reuse the common baseline models (cell grouping patterns, underlying feature groups), to analyze newly acquired images, thereby saving computing time. We describe this "batch learning" style of applying our method to experimental studies in this paper.

We present examples of its usefulness by analyzing synthetic datasets, as well as other experimental data such as brain perturbation study, binge alcohol study. We also tested our method for the benchmark dataset, cell reconstruction ensembles downloaded from the NeuroMorpho database. The proposed co-clustering and comparative interpretation algorithm is incorporated into an interactive graphical user interface around an intuitive heat map representation in the free and open-source FARSIGHT Trace Editor (Luisi et al., 2011, www.farsight-toolkit.org). We illustrate our comparative arbor analytics method from a neuro-engineering study aimed at profiling glial alterations resulting from the insertion of a neural recording device into the brain tissue.

The following chapters are organized as follows. In Chapter 2, we give a brief literature review for arbor analytics, co-clustering, dimensionality reduction and comparative analysis. Limited by the number of existing literatures on arbor analytics, we include a few of the recent ones in the section. Several types of co-clustering algorithms are described including two way clustering, matrix decomposing based clustering and graph based clustering. We also describe the major categories of dimensionality reduction methods including linear and non-linear approaches. We introduce our dimensionality reduction method (diffusion distance) in great detail in Chapter 3. The diffusion embedding requires data points to be connected as a weighted graph. A random walk probability field is constructed from the graph. We model the probability field as a Markov chain process so that the distance between data points is captured as the transition probability. In Chapter 4, we describe the details about the clustering algorithm that we are using in the co-clustering approach. It is a hierarchical clustering in the sense that it clusters points into clusters of different scales in different levels. The hierarchical structure is captures in a partition tree. The clustering procedure is a basic step for the co-clustering framework. The details and theoretical support of the co-clustering frame work is given in Chapter 5. First of all, we introduced a scheme that enhances the dual geometry in the co-clustering problem, to bring stronger affinity to the data points in the other space (row space or column space). The iterative procedure is described afterwards as the major steps of the method. Finally, we give the theory of orthogonal basis and smoothness as a convergence criterion, and a better explanation of the algorithm itself. We introduce the comparative self-interpretation algorithm in Chapter 6. The algorithm is formed based on sparse group lasso, in which a convex optimization objective function is formularized with regularizing both in the individual level and in a group wise level. A brief introduction about lasso and group lasso is given in the chapter, as well as the detail of the sparse group lasso algorithm. Chapter 7 includes the analytical results we have done on multiple experiments. One examples is the Microglial profiling in the rat brain perturbation experiment, where four sub status of microglial and their space distribution are identified. Another example is astrocyte profiling in the binge alcohol study, where astrocytes are profiled into four groups with different morphological characteristics. We also did analysis for neuron profiling, where different types of neurons, as well as neuron subtypes are identified. Finally, we applied the profiling algorithm and the interpretation algorithm on a binge alcohol study.

Chapter 2 Background and Prior Literature

2.1 Arbor Analytics

Quantitative analysis of arbor morphology data for neurons, microglia, and astrocytes has been of long-term interest to neuroscience, and simpler methods that just focus on a few arbor measurements computed on individual cells, or small populations of cells. For example, Vargas-Irwin et al., (2014) proposed a framework for single neuron and ensemble data analysis. An automated method to quantify microglia morphology and application to monitor activation state longitudinally in vivo was presented by Kozlowski et al., (2012). In Tan's work (2014), they analyzed the influence of gold surface texture on microglia morphology and activation.

The advent of comprehensive collections of arbor measurements, notably Ascoli's L-measure (Scorcioni et al., 2008; Parekh, et al., 2013), has set the stage for a more comprehensive style of quantitative arbor analysis. In this approach, manually/computationally reconstructed arbors are exported to a standardized data format, for example, the open SWC data representation (Cannon et al., 1998), and then automated computer programs are used to compute a comprehensive array of arbor measurements. This approach provides the opportunity for a far more extensive analysis without additional data acquisition. The potential offered by such analysis is tempered by the need for tools to handle the high data dimensionality.

Increasingly, these studies driven by growing collections of reconstructions (e.g., www.NeuroMorpho.org), and our growing ability to reconstruct arbors with a high degree of automation (e.g., Zhou et al., 2016, Peng et al. 2015, Hogrebe et al., 2012, Meijering 2010, Wang et al., 2011, Luisi et al., 2011, Peng et al., 2011, Turetken et al., 2011), reconstruct entire populations of arbors (Megjhani et al., 2015, Kulkarni et al., 2015, Rey-Villamizar et al., 2014), and compute widely accepted arbor measurement collections (Scorcioni et al., 2008, Parekh, et al., 2013). Our work exemplifies this recent trend. Specifically, this dissertation represents a much needed extension of our published paper (Lu et al., 2015), where we presented a robust unsupervised harmonic co-clustering method for profiling arbor morphologies for ensembles of reconstructed brain cells based on quantitative measurements of the cellular arbors. This method can identify groups and sub-groups of cells with similar arbor morphologies, and simultaneously identify the hierarchical grouping patterns among the quantitative arbor measurements. The more recent work of Xu et al., (2016) proposed a non-parametric Bayesian approach to unsupervised quantitative profiling of microglial arbor states, and mapping their 3-D spatial distributions across extended brain tissue regions imaged by mosaiced confocal microscopy. The arbor morphology data are fitted into an Infinite Gaussian Mixture Model (IGMM) with collapsed Gibbs sampling to discover arbor-morphological classes, their morphological ‘signatures,’ and the underlying covariance matrices, all in an unsupervised manner. Although these prior methods provide a significant level of feature grouping and analysis, they do so at the level of individual datasets. This paper provides the much-needed extension for comparative analysis of two or more data sets, with a systematic biologically meaningful strategy for interpreting the comparative arbor analytics results.

Within the realm of comparative analysis, the main contribution of this work is the identification of the most significant alterations across distinct groups. In the recent literature, Polavaram et al., 2014 presented a database-wide statistical analysis of dendritic arbors, enabling quantification of the major morphological similarities and differences across broadly adopted metadata categories. They also adopted a complementary unsupervised approach based on clustering and dimensionality reduction to identify the main morphological parameters leading to the most statistically informative structural classification. The purpose of this method is very similar to ours, except the approach is different, which leads to a different insight discovery. In our model, we derive the arbor morphological similarity/difference after the clustering analysis so we focus on comparison among groups. Also, the use of group LASSO in our model allows us to relate the alterations directly to the original arbor measurements and their biological functionality.

2.2 Co-clustering

The problem of co-clustering, where both rows and columns of a data matrix are clustered simultaneously, has received significant attention in bio-informatics, and other data-intensive areas of research. Dhillon & Mallela (2003), Cheng & Church (2000), George & Merugu (2005) are the pioneers applied co-clustering methods to gene expression data, text mining (document classification), and recommendation systems. Many co-clustering approaches have been described, including the hierarchical model (Xu et al. 2006), bi-clustering model (Cheng & Church 2000), pattern-based model (Hartigan 1972), and matrix decomposition model, etc.

2.1.1 Iterative Methods

In Chen and Church's algorithm, the goal is to find biclusters with low variance in a greedy way. It iteratively remove rows and columns with high residuals (or variances) out from the whole matrix. The iterative removing process stops when a certain criterion is satisfied by a given threshold. The criterion for achieve constant biclusters is based on the definition of a mean square residue (MSR)

$$\mathrm{MSR} = \frac{1}{|I||J|} \sum\nolimits_{i\in I, j\in J} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2, \tag{1}$$

where a_{ij} is the data element at row i and column j, a_{iJ}, a_{Ij} and a_{IJ} are the mean of the expression values in of row i, column j, and the whole bicluster respectively, for $i \in I$ and $j \in J$. For seeking constant row clusters, column clusters, or rectangle biclusters, MSR is an effective measurement. However, this metric is not suitable for scale and shift-scale biclusters.

BiMex is a co-clustering algorithm designed for gene expression like data, where only upregulate and downregulate biclusters are found. The algorithm works for binary data due to its mechanism that it seeks a rectangle bicluster with 1s in a matrix. If a given matrix is continues non-binary data, it should be normalized into to a binary matrix (with only elements 1s and 0s) before applying the algorithm. This is a divide and conquer algorithm starts with the whole matrix, iteratively dividing into checker board format.

Another algorithms works with manipulating matrix entry order is the Order Preserving Submatrix problem (OPSM) (Gao et al., 2006). In the OPSM model, a bicluster is defined as a submatrix within which the orders of entries are preserved. In each bicluster, there exists a linear ordering of the column in which the values of all rows within the bicluster is linearly increasing, and the other way, too (there exists a linear ordering of the row in which the values of all columns within the bicluster is linearly increasing). The algorithm constructs biclusters by iteratively increase the size of the partial biclusters, and ordering the probability of the submatrix reaching a fixed size defined by users. This algorithm is greedy and heuristic.

2.1.2 Matrix Decomposition Methods

Another important application for co-clustering algorithms is to find the hidden block structure of a data matrix. There are several typical methods proposed in the past decade dealing with this problem using matrix decomposition methods, such as Block Value Decomposition (BVD) and Non-negative Matrix Factorization (NMF).

Long et al., (2005) presented a co-clustering framework, the block value decomposition for dyadic data, which factorizes the dyadic data matrix into three components, the row-coefficient matrix R, the block value matrix B, and the column-coefficient matrix C. The coefficients denote the degrees of the rows and columns associated with their clusters and the block value matrix is an explicit and compact representation of the hidden block structure of the data matrix. Under this framework, they focused on a special yet very popular case – non-negative dyadic data, and propose a specific novel co-clustering algorithm that iteratively computes the three decomposition matrices based on the multiplicative updating rules.

In this method, data are formulated as dyadic data denoted as (x, y), given two sets $\boldsymbol{X} = \{x_1, x_2, \dots, x_n\}$ and $\boldsymbol{Y} = \{y_1, y_2, \dots, y_n\}$. For the scalar dyads, the data can always be organized as an $n \times m$ two-dimensional matrix $\boldsymbol{Z}$ by mapping the row indices into $\boldsymbol{X}$ and the column indices into $\boldsymbol{Y}$. Then, each *w(x, y)* corresponds to one element of $\boldsymbol{Z}$. The goal is to simultaneously clustering $\boldsymbol{X}$ into k disjoint clusters and $\boldsymbol{Y}$ into l disjoint clusters. This is equivalent to finding block structure of the matrix $\boldsymbol{Z}$, i.e., finding $k \times l$ submatrices of $\boldsymbol{Z}$

such that the elements within each submatrix are similar to each other and elements from different submatrices are dissimilar to each other. Since the elements within each block are similar to each other, we expect one center to represent each block. Therefore a $k \times l$ small matrix is considered as the compact representation for the original data matrix with a $k \times l$ block structure. In the traditional one-way clustering, given the cluster centers and the weights that denote degrees of observations associated with their clusters, one can approximate the original data by linear combinations of the cluster centers.

Non-negative matrix factorization (NMF) is a recent method for finding two low-rank non-negative matrices whose product provides a good approximation to the original nonnegative matrix. Given a data matrix $\boldsymbol{X} \in R^{M \times N}$, NMF aims to find two low-rank non-negative factors $\boldsymbol{U} \in R^{M \times R}$ and $\boldsymbol{V} \in R^{N \times R}$, which are the basis factor and low-rank representation factor, respectively. The squared error (Euclidean distance) objective function is formulated as

$$\mathrm{J}_{NMF} = \sum_{i=1}^{N} \sum_{j=1}^{M} (\mathrm{X}_{ij} - (UV^T)_{ij})^2 = \|X - UV^T\|_F^2 \tag{2}$$

subject to $U \geq 0, V \geq 0$.

In Shang et al., work (2012), a co-clustering method was proposed using graph dual regularization non-negative matrix factorization (DNMF) algorithm which simultaneously considers the geometric structure information contained in data points as well as features. They first construct a k-nearest neighbor data graph whose vertices correspond to $\{x_1, x_2, \dots, x_n\}$. And use the 0–1 weighting scheme for constructing the k-nearest neighbor graph as, and define the data weight matrix as

$$\mathrm{W}_{ij} = \begin{Bmatrix} 1, if\, \mathrm{X}_{ij} \in N(\mathrm{X}_i) \\ 0, otherwise \end{Bmatrix}, i, j = 1, \dots n, \tag{3}$$

where $N(\mathrm{X}_i)$ represent the set of k-nearest neighbors of X_i. The graph Laplacian of the data graph is defined as $L = D - W$, where D is a diagonal degree matrix whose entries are given by $\mathrm{D}_{ii} = \sum_j \mathrm{W}_{ij}$. Based on two graph regularizers of both data manifold and feature manifold, they proposed a novel graph dual regularization non-negative matrix factorization, whose objective function is formulated as

$$\mathrm{J}_{NMF} = \sum_{i=1}^{N} \sum_{j=1}^{M} (\mathrm{X}_{ij} - (UV^T)_{ij})^2 + \lambda Tr(V^T LV) + \mu Tr(U^T LU) \tag{4}$$

subject to $U \geq 0, V \geq 0$.

2.1.3 Others

The partitioning-based model, first introduced by Xu et al., (2006), has attracted much interest, because of the simplicity of the formulation, and its close relationships with other well studied problems, such as spectral clustering and matrix decomposition (Banerjee et al.. 2004; Ding et al.. 2005). Specifically, we also infer clusters of rows, clusters of columns, in addition to the coherent sub-matrix. The Coupled Two Way Clustering (CTWC) algorithm, introduced by Getz et al.. (2003), separately clusters the rows and columns at first, and then obtains a stable bi-cluster by iteratively combining pairs of row and column clusters. The Interrelated Two Way Clustering (ITWC) method proposed by Tang & Zhang, processes the data matrix by initially clustering on one space (either row space or column space), and then clustering on the other space within the initial clusters (Tang & Zhang 2005). Banerjee et al., (2004) proposed a generalized Bregman co-clustering algorithm by posing co-clustering as a matrix approximation problem. In our algorithm, we use matrix approximation as a criterion of how smooth the reorganized data is. Brameier & Wiuf (2007) presented a co-clustering approach based on Self-Organizing Maps (SOMs). SOMs are a probabilistic clustering method that imposes a neighborhood structure on the clusters. It combines the center-based clustering of standard SOMs with a representative-based clustering.

Most of the algorithms in the literature have been shown to perform well on small datasets in their respective application areas. However, a direct application of these methods to large high-dimensional neuro-morphological datasets proved impractical for several reasons. First, most clustering methods require the user to specify the number of clusters, and this is impractical when performing exploratory data analysis. Second, we found that as the population size grows, the computational times increased drastically, making interactive analysis impractical. Third, we found that the prior hierarchical clustering methods were overly sensitive to outliers and noise in the data. Specification of cluster number is not required in our method because the clustering is conducted in a multi-scale manner. Our algorithm is able to infer the hierarchical geometry of the row and column spaces, and also take into consideration the block structure of the entire data matrix simultaneously. The algorithm presented here, and its 64-bit implementation are designed to be scalable to large datasets containing large populations of cells (tens of thousands of cells or more), limited only by the speed and memory of the computer. Moreover, the proposed algorithm uses the diffusion distance (Coifman & Lafon 2006) as the distance metric (described further below), making it inherently robust to missing values and outliers due to image processing and automated tracing errors.

2.3 Dimensionality Reduction

In real world data mining applications, data dimensionality is usually very high due to the amount of available feature measuring ability. In order to handle and study the data better, it is very essential to lower the dimensionality of the data so that a compact and intrinsic structure is revealed. The dimensionality is of crucial importance especially for tasks like classification and clustering. The ability of finding out the intrinsic dimensionality and properties is directly limiting the performance of the following procedures. In this section, we review the current available dimensionality reduction algorithms in a systematic way. Dimensionality reduction algorithms can be classified as linear and nonlinear approaches based on the data structure the algorithm is able to handle. PCA is the most commonly used linear approach. While there are multiple categories of nonlinear approach. Nonlinear approach can be divided into three algorithms that preserve global properties, preserve local properties and alignments of local properties. There are several representative algorithms from each category that we are about to introduce in greater detail. The relationship between those algorithms are related to each other and even identical under certain circumstances.

The problem of dimensionality can be summarized as follows. Assume we have a data representation $\boldsymbol{X} \in \mathbb{R}^{n\times M}$**,** consisting of n data vectors $\boldsymbol{x}_i$, $i = 1,2,3, \dots, n$ with dimensionality D. Assume further that the data has an intrinsic dimensionality of d, where $d \leq D$. Dimensionality reduction techniques transform the data $\boldsymbol{X}$ into a new dataset $\boldsymbol{Y}$ with dimensionality d, while retain the fundamental geometry of the original dataset. Generally neither the intrinsic dimensionality d nor the preserved geometry is known or given for the problem. We briefly describe the algorithm and theory of a few popular state of art techniques in the following section.

2.2.1 Linear Dimensionality Reduction

Applying linear dimensionality reduction algorithm, the original high dimensional data is usually embedded into a lower dimensional space by a linear projection or mapping. Among the variety of linear methods, PCA is the most straightforward and robust method for such task. It is an unsupervised algorithm requires no input information and parameters.

PCA, Principal Components Analysis projects a high dimensional data into a lower space by preserving as much variances as presented in the original data. PCA finds a set of new basis from the original data, with maximal variances along each new basis. Embedded data is simply coordination's in the new basis coordination system. Assume the mapping function denoted as M, the goal is to find the mapping function M such that it maximize the covariance $M^T cov(\boldsymbol{X})M$ of the mapped data. Notice that $cov(\boldsymbol{X})$ is the covariance of the original data $\boldsymbol{X}$. Mathematically, it is straightforward that the new basis are the principle eigenvectors of the covariance matrix. Therefore, PCA solves the eigen system of the covariance of the original data $\boldsymbol{X}$

$$cov(\boldsymbol{X})M = \lambda M. \tag{5}$$

The eigen problem is solved for the d principal eigen values λ. The low-dimensional data representations y_i of the data points x_i are computed by mapping them onto the linear basis M, i.e., $\boldsymbol{Y} = (\boldsymbol{X} - \widetilde{\boldsymbol{X}})M$.

PCA has been specifically applied in multiple domain where dimensionality reduction is required. However, it can be time consuming since that the covariance matrix is proportional to the data dimension. Beside, solving the eigen problem is time consuming too. Therefore, in datasets where $n < D$, instead of computing the eigen system of the covariance system, we use the Euclidean distance matrix $(\boldsymbol{X} - \widetilde{\boldsymbol{X}})(\boldsymbol{X} - \widetilde{\boldsymbol{X}})^{\boldsymbol{T}}$ as an alternative.

2.2.2 Nonlinear Dimensionality Reduction

Among the nonlinear dimensionality reduction methods, there are mainly three categories based on the preservation of global or local relativity and connectivity. There are methods that preserve the global properties and connectivity of the original data in the embedded lower dimensional space. The most popular and often used global nonlinear techniques include multidimensional scaling (MDS), Isomap, and maximum variance unfolding (MVU), kernel PCA, and multilayer autoencoders. There are methods that preserve the local properties and connectivity of the original data in the embedded lower dimensional space. Local nonlinear techniques that are deployed in a lot of applications including local linear embedding (LLE), Laplacian eigen maps, Hessian local linear embedding (Hessian LLE), and local tangent space analysis (LTSA). And there are methods that preserve global alignment with a mixture of linear models. The two most famous global alignment techniques are local linear coordination (LLC), and manifold charting. We describe one technique from each subtype in the following paragraphs.

ISOMAP

Isomap is a dimensionality reduction method that preserves the global relativity and connectivity of the original data.

In isomap, the distance metric used is the geodesic distance between the data points $x_i, (i = 1,2,3, \dots, n)$. Geodesic distance is the distance between two points measured over the manifold. It is computed by constructing a neighborhood graph G, in which each every data point serves as a node, and the edges between nodes are only connected if it is within the k nearest neighbors $x_{ij}, (j = 1,2,3, \dots, n)$ in the dataset $\boldsymbol{X}$. Unlike in multidimensional scaling or similar techniques, where the Euclidean distance is applied, the density and distribution of the neighborhood is not taken into consideration. When data points lies in a Swiss roll like configuration, the two data points that are considered neighbors based on the Euclidean distance is not actually neighbors considering the local connectivity. In isomap, their distance over the manifold is much larger than appears. isomap preserves the pairwise locality by using geodesic or curvilinear distances between data points.

The geodesic distance between two points are estimated by using the shortest path found in the constructed graph, Dijkstra's is a well-known algorithm for solving such problems with elegancy and simplicity. Thus, the isomap computes a pairwise geodesic distance matrix between all data points from the original data matrix $\boldsymbol{X}$. After deriving the geodesic matrix, the embedding is achieved by applying multidimensional scaling on the resulting distance matrix. Therefore, the lower dimensional data y_i is mapped using MDS on x_i based on the geodesic distance, instead of the Euclidean distance.

Although Isomap has been successfully applied in variety of methods and applications, especially data visualization and bio-medical experiments, yet there are a few drawbacks with its topological instability. In building the neighborhood graph, there may be erroneous connections between data points which can lead to impact to the performance of the algorithm. When the manifold is nonconvex or "hole" formed, the Isomap might fail to extract the intricacy underling structure of the data. However, there are methods been proposed dealing with this problem. An example for overcoming the short circuiting is to removing data points with large total flows in the shortest path algorithm. Also it is compromised by removing the nearest neighbors that violate local linearity of the neighborhood graph.

Local Linear Embedding (LLE)

Local Linear Embedding (LLE) is a nonlinear dimensionality reduction method that preserves the local relativity and connectivity of the original data in the lower dimensional space.

Similarly, LLE derives data connectivity through constructing a neighborhood graph. The local properties of a data point x_i in the manifold is described by forming the data point as a linear combination W_i (the so-called reconstruction weights) of its k nearest neighbors x_{ij} . The assumption for locality preserving is: if the low-dimensional data representation preserves the local geometry of the manifold, the reconstruction weights W_i that reconstruct data point x_i from its neighbors in the high-dimensional data representation also reconstruct data point y_i from its neighbors in the low-dimensional data representation. Therefore, the objective is formulated as to minimize the cost function

$$\phi(\mathrm{Y}) = \sum_{i}(y_i - \sum_{j=1}^{k} w_{ij}y_{ij})^2. \tag{6}$$

For finding the d-dimensional data representation ***Y*.** It can be shown that the coordinates of the low dimensional representations y_i that minimize this cost function can be found by computing the eigenvectors corresponding to the smallest d nonzero eigenvalues of the inner product $(I - W)^T (I - W)$. In this formula, I is the $n \times n$ identity matrix.

It also implies that the reconstruction weight of the data point are invariant to translation, rotation and rescaling. Because of the invariance to these transformations, any linear mapping of the hyperplane to a space of lower dimensionality preserves the reconstruction weights in the space of lower dimensionality. Hence, LLE fits a hyperplane through the data point x_i and its nearest neighbors, thereby assuming that the manifold is locally linear.

The popularity of LLE has led to the proposal of linear variants of the algorithm, and to successful applications to, e.g., super resolution and sound source localization. However, there also exist experimental studies that report weak performance of LLE. A possible explanation lies in the difficulties that LLE has when confronted with manifolds that contains holes. In addition, LLE tends to collapse large portions of the data onto a single point in cases where the target dimensionality is too low.

Compared to Isomap, LLE attempts to preserve solely local properties of the data. As a result, LLE is less sensitive to short-circuiting than Isomap, because only a small number of properties are affected if short-circuiting occurs. Furthermore, the preservation of local properties allows for successful embedding of nonconvex manifolds. In LLE, the local properties of the data manifold are constructed by writing the data points as a linear combination of their nearest neighbors. In the low-dimensional representation of the data, LLE attempts to retain the reconstruction weights in the linear combinations as good as possible (Roweis, et al., 2000).

Local Linear Coordination (LLC)

Locally Linear Coordination (LLC) is dimensionality reduction method that preserve the global alignments with a mixture of linear models. The procedure of LLC consists two stages, the mixture stage and the alignment stage. In the mixture stage, LLC computes a mixture of local linear models in the original data space with methods like EM. While in the alignment stage, the local modes are aligned in order to obtain the low dimensional data representation using a variant of LLE.

LLC first constructs a Mixture of m Factor Analyzers (MoFA) using the EM algorithm. Alternatively, a Mixture of Probabilistic PCA models (MoPPCA) model could be employed. The local linear models in the mixture output m data representations z_{ij} and their corresponding responsibilities r_{ij}, where $j = 1,2,3, \dots, m,$ for every data point x_i. The responsibilities r_{ij} describe to what extent data point x_i corresponds to the model j, they satisfy$\sum_j r_{ij} = 1$. Using the local models and the corresponding responsibilities, responsibility-weighted data representations $u_{ij} = r_{ij}z_{ij}$ are computed. The responsibility-weighted data representations u_{ij} are stored in an $n \times m$ D block matrix $\boldsymbol{U}$. The alignment of the local models is performed based on $\boldsymbol{U}$ and on a matrix $\boldsymbol{M}$ that is given by $\boldsymbol{M = (I - W)^T (I - W)}$. Herein, the matrix $\boldsymbol{W}$ contains the reconstruction weights computed by LLE, and $\boldsymbol{I}$ denotes the $n \times n$ identity matrix. LLC aligns the local models by solving the generalized eigen problem

$$\boldsymbol{A}v = \lambda \boldsymbol{B}v \tag{7}$$

for the d smallest nonzero eigenvalues. In the equation, A is the inner product of $\boldsymbol{M}^T\boldsymbol{U}$ and $\boldsymbol{B}$ is the inner product of $\boldsymbol{U}$. The d eigenvectors v_i form a matrix L, that can be shown to define a linear mapping from the responsibility-weighted data representation $\boldsymbol{U}$ to the underlying low-dimensional data representation $\boldsymbol{Y}$. The low-dimensional data representation is thus obtained by computing $\boldsymbol{Y = UL}$.

2.4 Sparse Regularization

In this dissertation, in order to achieve an automatic way of discovering the significant alterations, we propose a comparative analysis model based on an adaptation of the sparse group LASSO algorithm (Ming et al., 2006, Simon et al., 2013). The group lasso was first proposed by Ming et al., (2006). In their work, they considered the problem of selecting grouped variables (factors) for accurate prediction in regression. Such a problem arises naturally in many practical situations with the multifactor analysis-of-variance problem as the most important and well-known example. Instead of selecting factors by stepwise backward elimination, they focused on the accuracy of estimation and considered extensions of the LASSO, the LARS algorithm (Efron, et al., 2004) and the non-negative garrote for factor selection (Yuan, 2007). For high-dimensional supervised learning problems, using problem-specific assumptions can sometimes lead to greater accuracy. Simon et al., (2013) presented a sparse group lasso algorithm in their work. They introduced a regularized model for linear regression with ℓ1 and ℓ2 penalties and showed that it has the desired properties of group-wise and within-group sparsity. They proposed the algorithm to fit the model via an accelerated generalized gradient descent method, and extended the model and algorithm to convex loss functions. Our comparative interpretation algorithm adopts the sparse group LASSO, using a predefined group of L-measure features, and the output yields a ranking of the significant feature groups and their significance scores. In other words, it gives a robust ranking of the alterations in terms of the feature groups.

As noted above, the practical case when batches (or a series) of images are being acquired under a common imaging protocol, typically as part of a larger ongoing study, represent an important use case for comparative arbor analytics. The proposed method is readily usable in a "batch learning" style in which mathematical models extracted from a representative subset of the data can be reused for processing datasets from similar experiments and applications. This idea is not new, and similar concepts using an unsupervised learned model for faster processing was proposed earlier. For example, in the exploratory environment learning application (Amershi et al., 2009), they presented a data based user modeling framework that uses both unsupervised and supervised classification approaches to build student models for exploratory learning environments. They applied the framework to build student models for two different learning environments and using two different data sources (logged interface and eye-tracking data). Despite some limitations due to the size of datasets, they have provided initial evidence that the framework can automatically identify meaningful student interaction behaviors and can be used to build user models for the online classification of new student behaviors online. Kyriakopoulou et al., (2006) proposed a text classification model using clustering in their work. They addressed the problem of learning to classify texts by exploiting information derived from both training and testing sets. To accomplish this, clustering is used as a complementary step to text classification, and is applied not only to the training set but also to the testing set.

Chapter 3 Diffusion Embedding and Diffusion Distance

In practical applications, it is usually important and necessary to analyzing the underlying geometric structure of the given dataset, as well as the local and global correlation between data points. Given a data matrix $\boldsymbol{X} \in \mathbb{R}^{m\times n}$,say the rows of which are observations, and the columns of which are variables (features), the task of discovering data structure and correlation is becoming extremely difficult and unreliable in high dimensional space using conventional methods. Therefore, it is essential to reduce the dimensionality of the data, and represent the data in another space while keeping the relative structure of the dataset.

In traditional methods, the correlation between data points is often measured by a similarity/distance function. For example, the distance described in Euclidean space

$$d(\boldsymbol{x}_i,\boldsymbol{x}_j) = \left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_2 = \sqrt[2]{\sum_{z=1}^{n}(x_{iz} - x_{jz})^2}. \qquad (8)$$

However, as the dimensionality increases, it is unreliable to capture the connectivity between data points. The global structure constructed is highly biased by the high dimensionality. Besides, the distance function is sensitive to noise given the correlation is computed between single pair of data points.

In this dissertation, we propose a new method of reducing the data dimensionality using the diffusion theory. In our method, we represent our data in the embedded lower dimensional space. The lower dimensional space is achieved by sampling a forward Markov chain. The Markov chain is achieved from a random walk field constructed based on a connected graph. We first construct a connected graph on the datasets, the vertices in which represents data points, and the edges in which represent affinities between data points. The second step is to construct a probability field from the connected graph. The probability of walking from one data point to another is defined as the affinity between the two points normalized by the local volume (density). This random probability field satisfies two conditions: static and reversible, thus we can construct a Markov chain from it. Running forward the chain with different time steps yields multi-scales analysis on the original dataset (will be described further). Mathematically, this leads to the graph Laplacian, and the computation can be done using spectral analysis. The diffusion distance therefore can be defined both in the probability field as well as in the embedded lower dimensional space. The distances computed this way run over the entire dataset with considering all connections, therefore, it is highly robust to random noises appear in the dataset.

3.1 Graph and Sparse Affinity

Graphs can be used to model many types of relations and processes in physical, biological, social and information systems. It is a mathematical structure used to model pair wise relation between objects. A weighted graph is composed of vertices and edges, $\boldsymbol{G}(\boldsymbol{V},\boldsymbol{E})$, where the set of vertices $\boldsymbol{V}$ represents objects, while the set of edges $\boldsymbol{E}$ represents affinity between objects. In an undirected graph, the edge between two objects has no orientation. In other words, $e(a,b) = e(b,a)$. The graph captures both the local connectivity and the global activity of the network.

In our method, we construct a graph network on the data set. Take $\boldsymbol{X} \in \mathbb{R}^{m\times n}$ as an example, each observation $\boldsymbol{x}_i$is treated as a vertex in the graph, the edges between vertices $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is assigned as the affinity value computed using an affinity kernel. The graph is described and stored as a matrix $\boldsymbol{G} \in \mathbb{R}^{m \times m}$, the entry $g(i,j)$ stores the edge $e(\boldsymbol{x}_i, \boldsymbol{x}_j)$. The affinity kernel $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$: $\boldsymbol{x} \times \boldsymbol{x} \overset{\Delta}{\rightarrow} \mathbb{R}$ has to satisfies two conditions:

- k is symmetric: $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = k(\boldsymbol{x}_j, \boldsymbol{x}_i)$,
- k is positivity preserving: $k(\boldsymbol{x}_j, \boldsymbol{x}_i) \geq 0$.

The selection of affinity kernel can be tricky. In most conditions, the affinity kernel should be able to capture the connectivity both locally and globally. But in our case, the diffusion theory applied later requires the local information to be preserved as much as possible, while the unreliable global information can be discarded accordingly. Therefore, it is essential to select affinity kernels that enhance the nearest neighbors' connectivity. In our case, we choose sparse affinity kernel, which resulted in affinity values only among neighboring data points, the affinity between far away data points are zeros. This yields a sparse graph. It is schematically illustrated in Figure 1. In Figure 1(a), the graph keeps only neighboring points connected, and it yields sub-graphs already without further analysis. While in Figure 1(b), the graph keeps connectivity between all data points, when most of the weight is biased due to high dimensionality. The sparse affinity kernel is a modification of the Gaussian kernel

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{-\|x_i - x_j\|^{\alpha}/\varepsilon}. \tag{9}$$

Parameter α and ε control the decaying speed and scale of affinity values. In another word, the two parameters define how sparse the affinity matrix (the graph) is. If the parameter α and ε are carefully selected, the affinity matrix will achieve sparsely for the given dataset. The main purpose of using the sparse affinity is to preserve and emphasize

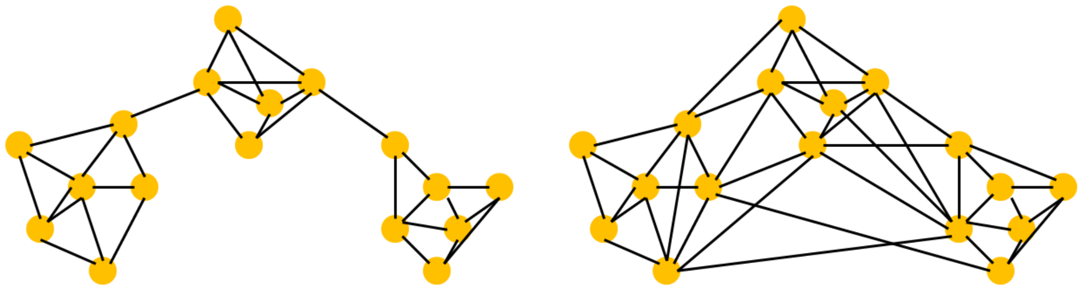


Figure 1. Constructed graph using different affinity criterions. The left graph is constructed by sparse affinity, only neighbors are maintained for further analysis. The right graph is constructed by conventional affinity function, it is fully connected.

the local information without bias. The global information will be discovered by further diffusion procedure.

3.2 Random Walk Field

Graph based random walk is a random walk on a weighted graph, where at each step the location jumps to another vertex according to some probability distribution associated with the edge weight. In a simple random walk, the location can only jump to neighboring vertices of the graph.

Before introducing the transition (jump) probability distribution, there are a few important concepts to be defined. Assume the graph is constructed as described above, the edge weights between data points are assigned with the affinities. The weights are positively related to the similarity between data points. The closer data points are in the variable space, the higher the weights are. If a data point $\boldsymbol{x}_i$ is close to many other data points in the neighborhood, the sum of the weights between this data point and other data points $\sum_{j\neq i}^{j\in\mathbb{N}} k(\boldsymbol{x}_i,\boldsymbol{x}_j)$ is high in general. Thus brings us to the definition of local density

$$d(\boldsymbol{x}_i) = \sum_{j\neq i}^{j\in\mathbb{N}} k\big(\boldsymbol{x}_i,\boldsymbol{x}_j\big), \tag{10}$$

where ℕ is the connected neighborhood. The transition probability is not only related to the affinity between the vertices, but also related to local density of the data points. Thus we define the transition probability as a function of

$$p(\boldsymbol{x}_i,\boldsymbol{x}_j) = \frac{k\big(\boldsymbol{x}_i,\boldsymbol{x}_j\big)}{d(\boldsymbol{x}_i)}. \tag{11}$$

An important fact of the transition probability is that it satisfies

$$\sum_{j\neq i}^{j\in\mathbb{N}} p(\boldsymbol{x}_i,\boldsymbol{x}_j) = 1. \tag{12}$$

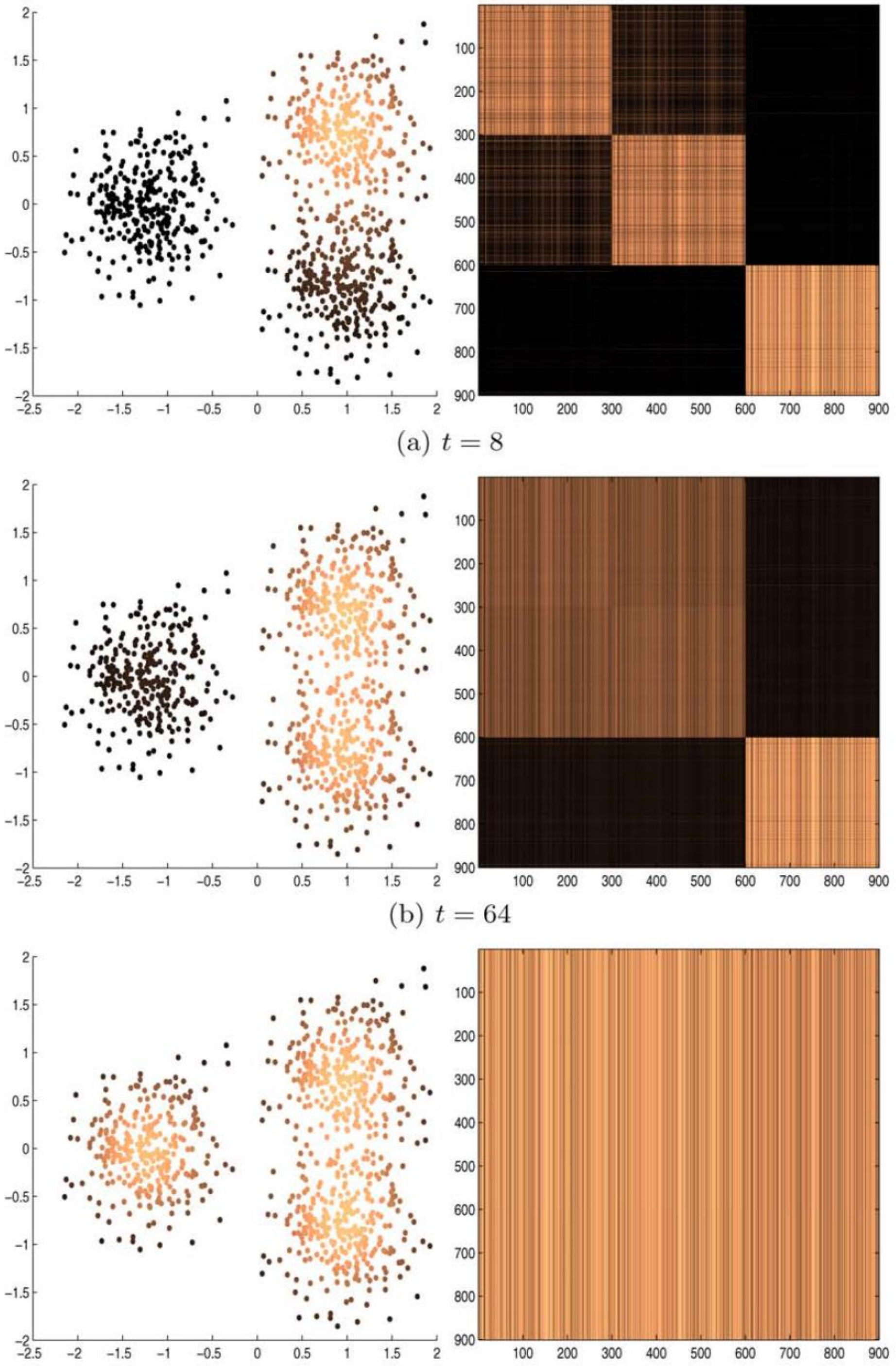
Based on the above equations, we define a transition matrix $\boldsymbol{M}$ on the random walk field, the entries $m_{i,j}$ is the transition probability $p\big(\boldsymbol{x}_i,\boldsymbol{x}_j\big)$. Mathematically, The transition matrix is computed by

$$\boldsymbol{M} = \boldsymbol{D}^{-1}\boldsymbol{G}, \tag{13}$$

where $\boldsymbol{D}$ is a diagonal matrix with elements $d(\boldsymbol{x}_i)$, and is the graph affinity matrix.

The above transition matrix defines the random walk on the dataset within one time step. If we run the chain forward in t time points, equally take the power of $\boldsymbol{M}$, $\boldsymbol{M_t} = \boldsymbol{M^t}$, it allows us to view the dataset in a different scales (large scales reveals the global geometry, while small scales reveals the local connectivity). From a data analysis point of view, large scales infer clusters with larger radius, and small scales infer cluster information with smaller radius. This is illustrated by the following example in Figure 2.

We generate a dataset of 1200 data points. The 1200 data points are randomly sampled from three 2-D Gaussian distribution. The centroids of the three distributions are $\{1, 0.8\}^T$, $\{1, -0.8\}^T$, and $\{-1.5, 0\}^T$. The covariance matrix for all three distributions is the unitary matrix. We form the transition matrix following the procedures above, and compute the power of the transition matrix. In the experiment, we run different time steps (different scales), three of them are shown in the figure, namely, $t = 8$, $t = 64$, and $t = 1024$. In the figure, the plot on the left side is the scatter plot of the data points coordinates, it is color coded by the transition probability at time t from one fixed point (sampled from the first distribution) to others. The matrix shown on the right hand side are the transition matrix at time t. The diffusion process at different time point reveals different cluster information as can been inferred from the figure. At $t = 8$, the diffusion process indicates a cluster of 3 with significant and clear boundary regarding the transition probability. At $t = 64$, the first two distributions are merged into one cluster, which grouping the dataset in a larger scale. While at $t = 1024$, the entire data set are integrated into one group. Therefore, the group information can be clearly identified with delicate selection of transition scales adapted to the data, as well as to the application. In the figure below, we show the transition matrix and diffusion process at different time t. On the left side, data sampled from three 2-D Gaussian distributions are plotted with their spatial coordinates. The data points are color coded by the transition probability from a fixed point. On the right side are the transition matrices at time $t = 8$, $t = 64$, and $t = 1024$ respectively.



(a) $t = 8$

(b) $t = 64$

(c) $t = 1024$

Figure 2. Transition matrix and diffusion process at different time t. On the left side, data sampled from three 2-D Gaussian distributions are plotted with their spatial coordinates. Data points are color coded by the transition probability.

3.3 Markov Chain and Spectral Analysis

A Markov chain (discrete-time Markov chain or DTMC), is a mathematical system that undergoes transitions from one state to another on a state space. It is a random process usually characterized as memory less: the next state depends only on the current state and not on the sequence of events that preceded it. A discrete-time random process involves a system which is in a certain state at each step, with the state changing randomly between steps. The steps are often thought of as moments in time, but they can equally well refer to physical distance or any other discrete measurement. Formally, the steps are the integers or natural numbers, and the random process is a mapping of these to states. The Markov property states that the conditional probability distribution for the system at the next step (and in fact at all future steps) depends only on the current state of the system, and not additionally on the state of the system at previous steps.

The random process we define above in previous section leads us to the Markov process given it satisfies the critical mathematical properties:

- The random walk satisfy the stationary property

$$p(\boldsymbol{x}) = \frac{d(\boldsymbol{x})}{\sum_{\boldsymbol{y} \in \boldsymbol{X}} d(\boldsymbol{y})}. \tag{14}$$

- The transition is reversible

$$p(\boldsymbol{x})p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{y})p(\boldsymbol{y}, \boldsymbol{x}). \tag{15}$$

- The random walk process is ergodic given the dataset $\boldsymbol{X}$ is finite.

The above construction of the Markov chain naturally leads us to the spectral analysis (analysis based on the Eigen system). Matrix $\boldsymbol{M} = \boldsymbol{D}^{-1}\boldsymbol{G}$ is not necessarily a symmetric matrix, but $\boldsymbol{G}$ is symmetric since it's the affinity between data points and $\boldsymbol{D}^{-1}$ is non-singular diagonal matrix. Therefore we have $\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{G}\boldsymbol{D}^{\frac{1}{2}}$ is symmetric. Then we have $\breve{\boldsymbol{M}} = \boldsymbol{D}^{-1}\boldsymbol{G} = \boldsymbol{D}^{-\frac{1}{2}}(\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{G}\boldsymbol{D}^{-\frac{1}{2}})\boldsymbol{D}^{\frac{1}{2}}$, it is easily seen that $\breve{\boldsymbol{M}}$ and $\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{G}\boldsymbol{D}^{-\frac{1}{2}}$ share the same eigen values and eigen vectors. Take the eigen values $\{\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_m\}$ and the eigen vectors $\{\psi_1, \psi_2, \psi_3, \dots, \psi_m\}$. We can define the diffusion distance and diffusion embedding in the next section.

3.4 Diffusion Distance and Diffusion Embedding

In Section 3.2 and 3.3, we defined the random walk and Markov chain on the data. In this section, we will introduce the distance measure of the original data in the embedded diffusion space and in the probability field. We interpret the diffusion distance in the probability field, while the computation of distance and embedding will be given based on the spectral analysis.

Within one time step, the probability of transition from $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$ is given by $p(\boldsymbol{x}_i,\ \boldsymbol{x_j})$. The probability describes the similarity of the data points in some extent. The fact of random noise could contribute a lot to such a single link measurement. Therefore, in our process, we define the similarity measurement not only concerning the pair of data points interested. We also consider the neighborhood connection of the two points. In another word, the connection from $\boldsymbol{x}_i$ to $\boldsymbol{y}$, where $\boldsymbol{y} \epsilon \mathbb{N}^{x_i}$($\boldsymbol{y}$ is a data point in $\boldsymbol{x}_i$ neighborhood), and the connection from $\boldsymbol{y}$ to $\boldsymbol{x}_i$ both contribute to the similarity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Hence, the likelihood of walking from $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$ within one time step is defined as

$$d(\boldsymbol{x}_i, \boldsymbol{x_j}) = \sum_{\boldsymbol{y} \neq \boldsymbol{x}_i}^{\boldsymbol{y} \epsilon \mathbb{N}^{x_i}} \left\| p(\boldsymbol{x}_i, \boldsymbol{y}) - p(\boldsymbol{y}, \boldsymbol{x}_j) \right\|_{\beta}. \tag{16}$$

We usually choose $\beta = 2$ to be the L-2 normal. In advance, we also define the diffusion distance over multiple time steps,

$$d_t(\boldsymbol{x}_i, \boldsymbol{x_j}) = \sum_{\boldsymbol{y} \neq \boldsymbol{x}_i}^{\boldsymbol{y} \epsilon \mathbb{N}^{x_i}} \left\| p_t(\boldsymbol{x}_i, \boldsymbol{y}) - p_t\big(\boldsymbol{y}, \boldsymbol{x}_j\ \big) \right\|_\beta \tag{17}$$

by taking the eigen values and eigen vectors, we map the original data point into the spectral space $\varphi_t : \Omega \overset{\Delta}{\rightarrow} \mathbb{R}^n$,

$$\varphi_t(x) = \{\lambda_1^t \psi_1(x),\ \lambda_2^t \psi_2(x),\ \lambda_3^t \psi_3(x), \dots, \lambda_n^t \psi_n(x)\}. \tag{18}$$

The eigen values are derived from the Markov process, therefore it is known to be $\|\lambda_i\| \leq 1$. In this process, we can choose the dimensionality of the embedded space as where the eigen value drop significantly (say s). The embedded diffusion space is then defines as

$$\Psi_t(\boldsymbol{X}) \triangleq \begin{pmatrix} \lambda_1^t \psi_1(x) \\ \lambda_2^t \psi_2(x) \\ \vdots \\ \lambda_s^t \psi_s(x) \end{pmatrix}. \tag{19}$$

Further we define the diffusion distance in the spectral space as

$$d_t\big(\boldsymbol{x}_i, \boldsymbol{x_j}\big) = \sum_{l \geq 0} \lambda_l^t \left\| \varphi_t(\boldsymbol{x}_i) - \varphi_t\big(\boldsymbol{x}_j\big)\ \right\|_\beta. \tag{20}$$

3.5 Examples

To demonstrate the ability of the diffusion embedding in dimensionality reduction and capturing global information among dataset, we created a synthetic dataset as well as a real word dataset. We also visualize the embedding results in the diffused space. The visualization is conducted on a 3-D embedded space for simplicity.

The first dataset created is a matrix with 256 rows and 128 columns. The entries of the matrix is a realization of the function $M_{i,j} = \frac{1}{2}(1 + \sin(\frac{i+j+2ij}{2}))$. The matrix is embedded into the diffusion space of both its row space and column space. The results are as shown in Figure 3. The embedded data reveals clear geometric structure of the original dataset, while the same interpreting is not possible in the original row and columns spaces. Though the embedded data does not appear a linear separable structure, it is very straightforward and manageable to use any simple kernel on top of it in order to separate the data into adapted groups (clusters) according to the application at hand.

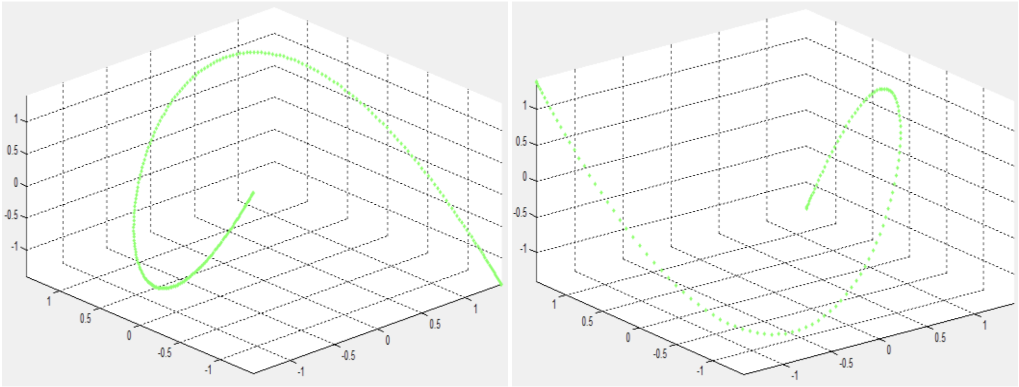


Figure 3. Embedding of a data matrix with 128*256 entries. The plot on the left is the embedding from the column space, the plot on the right side is the embedding from the row space.

We also applied the diffusion embedding on another data set with 2,418×567 entries. The data is created by the Minnesota Multiphase Psychological Inventory, which is the most widely used standardized psychological test. It is based on a survey with 567 relating questions among 2,418 people sampled. The data is only embedded from the observation space into the diffusion space. In the embedding space, the data is gathered in a relatively dense manner, but the boundary between different groups is easy to identify using any classifier. After construct the probability field and the Markov chain, the result is finally achieved at $t = 3$ in Figure 4.

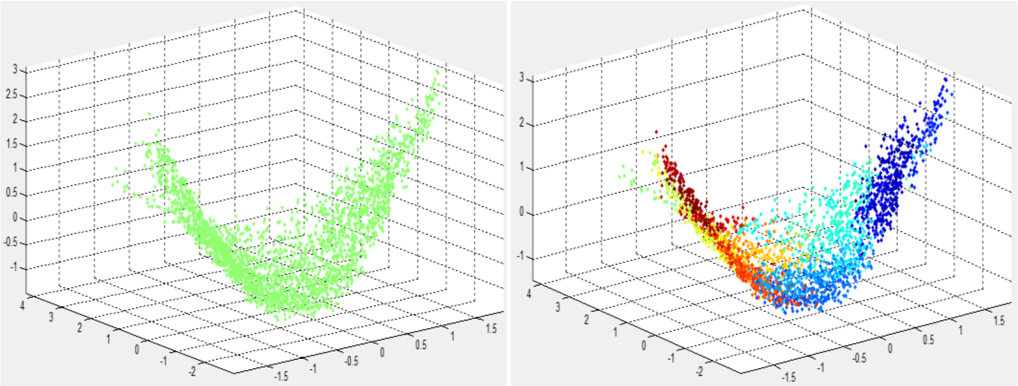


Figure 4. Embedding of a data matrix with 2,428×567 entries. The plot on the left is the embedding of the 2428 data points, the plot on the right side is the labeled embedding.

In both of the examples shown above, we choose the sparse affinity introduced $(k(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{-\|x_i - x_j\|^{\alpha}/\varepsilon}, \alpha = 2, \varepsilon = 0.7)$. For the computation of diffusion distance, we chose to use L-2 norm $(d_t(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{l\geq 0} \lambda_l^t \|\varphi_t(\boldsymbol{x}_i) - \varphi_t(\boldsymbol{x}_j)\|_{\beta}, \beta = 2)$.

Chapter 4 Hierarchical Partition Tree and Orthonormal Basis

In the area of data organizing, machine learning and pattern recognition, there are many ways to organize a dataset into a meaningful structure. In this chapter, we introduce a way of organizing a dataset into a hierarchical partition tree. The basic concepts about partition and hierarchical tree will be given later in this chapter. There are many formats of hierarchical tree can be constructed from a dataset. We introduce a classical binary tree adapted symmetrical data structures and flexible trees adapt to general data types as well. The goal of constructing a hierarchical partition tree on a dataset is to extract the group (cluster) structure in multiple levels and multiple scales, yet to have a good sight of how good the derived structure is. Therefore, we also introduce a set of orthogonal basis derived from the hierarchical partition tree. The orthogonal basis can be used for expansion of the original dataset, further being used as a criterion of measuring the smoothness of the original dataset.

This chapter is organized as follows: In Section 4.1, we introduce the basic concepts notations about partition and hierarchical tree. A brief introduction and comparison of binary tree and flexible tree is given in Section 4.2. In Section 4.3, we give several algorithms of how to build a hierarchical tree in both bottom up and top down procedures. An induction of an orthogonal basis and its theoretical support is given in Section 4.4. Finally, a brief summary is presented in Section 4.5.

4.1 Hierarchical Partition Tree

4.1.1 Basic Concepts

Partition is a division of a set. A partition of a set $\boldsymbol{X}$ is a set of non-empty subsets, such that each element $\boldsymbol{x}$ is in one subset and only in one subset. OR $\boldsymbol{X}$ is a disjoint union of the subsets. A set of subsets $P\ is$ a partition of set $\boldsymbol{X}$ only if it satisfy the following conditions:

- P does not contain any empty set, $\emptyset \notin P$.
- The union of the sets in P is equal to $\boldsymbol{X}$,$\bigcup_{p_i \in P} p_i = \boldsymbol{X}$.
- The intersection of any distinct sets in P is empty, if $p_i, p_j \in P, and\ p_i \neq p_j, then\ p_i \cap p_j = \emptyset$.

A tree structure or tree diagram is a way of representing the hierarchical nature of a structure in a graphical form. It is named a "tree structure" because the classic representation resembles a tree, even though the chart is generally upside down compared to an actual tree, with the "root" at the top and the "leaves" at the bottom. A hierarchical tree is a tree structure consists of multiple levels. Each level is consisting of elements called “node”. The bottom levels of the tree are the leaves and the top level of the tree is the root. All nodes in the internal level are connected to its upper and lower level through its children (lower level) and parent (upper level).

4.1.2 Hierarchical Partition Tree

Hierarchical partition tree is a crucial structure of representing and organizing data in our algorithms. In this section, we give the details of the representation of a hierarchical partition tree and its construction. In this tree structure, if each level is treated as a set, then each level is a partition of its lower level, as well as the original data set. The partition of each level is divided by varies of clustering algorithms, and the correlation between data points are determined in the embedded diffusion space as described earlier. A schematic illustration of a hierarchical partition tree is as shown in Figure 5.

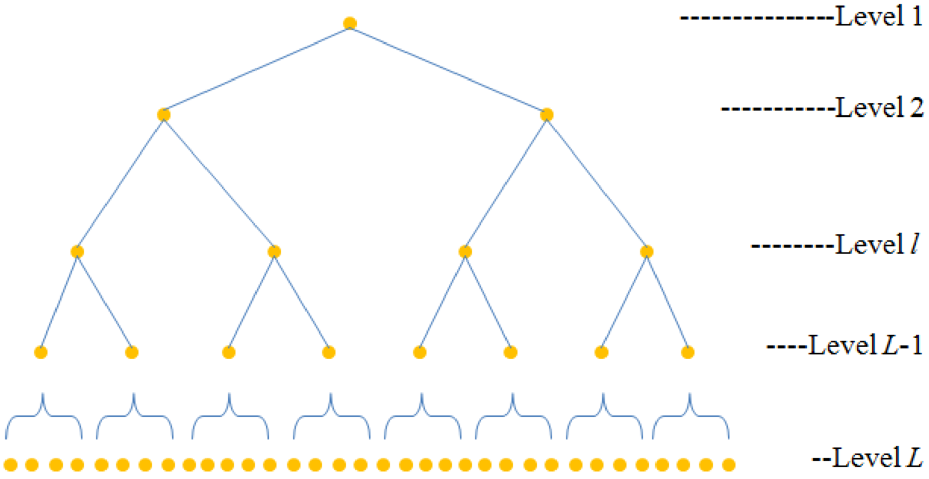


Figure 5. Hierarchical partition tree model

Assume again, a data set $\boldsymbol{X} \in \mathbb{R}^{m\times n}$ is given, we aim to organize the dataset in its feature space, namely we treat each of the $\boldsymbol{x}_i \in \mathbb{R}^n, i = 1,2,3, \dots, m$ as a data point. Say the final tree structure includes L levels. Thus we can define the bottom level with m elements in the original set (m leaves in the bottom level), denoted as

$$\boldsymbol{X} = \boldsymbol{X}^L = \bigcup_{i=1}^{m} \boldsymbol{x}_i = \bigcup_{i=1}^{m} \boldsymbol{n}^L{}_i. \tag{21}$$

In this level, $\boldsymbol{n}^L{}_i = \boldsymbol{x}_i$. The partition in this level is consists of m distinct subsets, each of the subset contains a single element from the original data set. To construct the $(L - 1)^{th}$ level (The second to the bottom level), we apply a certain clustering algorithm which will be given in later sections, so that the m nodes will be assigned in to K_{L-1}

groups (K_{L-1}denotes the number of nodes in the $(L-1)^{th}$ level). Therefore, the $(L-1)^{th}$ level is a partition of the L^{th} level. We denote the level as

$$\boldsymbol{X} \triangleq \boldsymbol{X}^{L-1} = \bigcup_{i=1}^{K_{L-1}} \boldsymbol{n}^{L-1}{}_{i}. \tag{22}$$

Similar but slightly different, $\boldsymbol{n}^{L}{}_{i}$ is a function of the node's children, $\boldsymbol{n}^{L}{}_{i} = f(\boldsymbol{n}^{L-1}{}_{1}, \boldsymbol{n}^{L-1}{}_{2}, \dots, \boldsymbol{n}^{L-1}{}_{J})$, where J is the size of the current node (number of children). The selection of clustering algorithm and the function $f = (.)$ can be various according to different dataset and applications, we will discuss later in greater details.

The same schema is used for building the $(l-1)^{th}$ level from the l^{th} level. We clustering the K_{l-1} elements into K_l groups, each nodes are represented as $\boldsymbol{n}^{l}{}_{i} = f(\boldsymbol{n}^{l-1}{}_{1}, \boldsymbol{n}^{l-1}{}_{2}, \dots, \boldsymbol{n}^{l-1}{}_{J})$, and the partition is denoted as

$$\boldsymbol{X} \triangleq \boldsymbol{X}^{l-1} = \bigcup_{i=1}^{K_{l-1}} \boldsymbol{n}^{l-1}{}_{i}. \tag{23}$$

The procedure is continued until there is only one group left, in another words, we have reach the root of tree. In summary the entire procedure can be denoted as

$$\begin{gathered}\boldsymbol{X} = \boldsymbol{X}^{L} = \bigcup_{i=1}^{m} \boldsymbol{x}_i \triangleq \boldsymbol{X}^{L-1} = \bigcup_{i=1}^{K_{L-1}} \boldsymbol{n}^{L-1}{}_{i} \boldsymbol{X} \triangleq \boldsymbol{X}^{l-1} = \bigcup_{i=1}^{K_{l-1}} \boldsymbol{n}^{l-1}{}_{i} \\ \triangleq \cdots \boldsymbol{X} \triangleq \boldsymbol{X}^{1} = \bigcup_{i=1}^{K_1} \boldsymbol{n}^{1}{}_{i}.\end{gathered} \tag{24}$$

4.2 Tree Construction

In the bottom-up approach, we describe the details of each step for constructing a flexible hierarchical tree. The hierarchical clustering we used in our method is based on the classic k-means algorithm. We modify the algorithm in different aspect to adapt to the current scenario. The algorithm is given and described as follows:

In the first step of this algorithm, each row of X is treated as a multivariate data point with dimension m. We assume that the structure of these data points can be modeled by a hierarchical partition tree denoted $T_r = \bigcup_{l=1}^{L} T_r^l$, where T_r^l is the lth level of the tree. This tree is constructed as follows. The lowest level, denoted T_r^L, is composed of n elements from the data matrix X. We group these n elements from T_r^L into K_{L-1}^r clusters. For this, the standard k-means algorithm is modified as follows:

a. The number of clusters at each level is chosen automatically as a fixed fraction of the number of items to be clustered. For the experiments reported here, we used 1/5 as the default setting, and this was chosen empirically. The intuition behind this setting is that we expect to cluster elements into groups that contain roughly 1/5 of the data points at each level. The user of this algorithm can adjust this setting based on this intuition.

b. We impose a constraint on the maximum cluster radii of the k-means algorithm to promote the formation of homogeneous clusters. The cluster radii are constrained to be less than a threshold. For the experiments reported here, we used a threshold value of 1.5 times the average distance of the intra-cluster data points to the cluster centroids. Increasing this setting beyond 1.5 can be expected to result in larger clusters being formed, and/or potential outliers being included within the clusters. If there are data points that exceed all of the cluster radii constraints, k-means clustering is performed with a larger number of initial partitions. The data are presented to the k-means algorithm in a consistent manner, in order to minimize the variability in the output due to the manner in which the data points are ordered. With this in mind, the cluster centroids for the k-means algorithm are initialized as follows. At each level at which the k-means algorithm is applied, we use a standard single-link agglomerative hierarchical clustering algorithm with a Euclidean distance measure to organize the data points into an ordered list in which similar data points (in the sense of the Euclidean distance measure) are consistently placed closer to each other. The number of cluster centroids for the k-means algorithm is known at each level as described above, and these cluster centroids are sampled uniformly from this ordered list of data points. One practical concern with large datasets is the time required to perform the hierarchical clustering. With this in mind, we split large datasets (containing more than 10,000 points) into several parts, with 2,000 data points per part, and sample evenly across these parts.

c. The clustering is driven by the diffusion distance metric instead of the traditional Euclidean distance measure (Coifman & Lafon 2006; Coifman & Maggioni 2006). In computing the diffusion distances, different t values are used at different levels, reflecting the fact that clustering is conducted at different scales. The scale of t is set through a fraction ε of similarity values such that $S_{i,j} = S_{j,i} = e^{-\|x_i - x_j\|^2/\varepsilon}$, where ε is initialized to $1/\sqrt{2}$ and scaled up by $\sqrt{2}$ for each higher level.

With this initial clustering, the centroids of the K_{L-1}^r clusters constitute the T_r^{L-1} level of the tree. We repeat this procedure by clustering the elements from T_r^{L-1} into K_{L-2}^r

clusters to form the T_r^{L-2} level. We continue clustering in this manner until there is only one all-inclusive cluster T_r^1.

The above-mentioned parameter settings are not altogether independent. For example, in choosing the fraction number (1/5) in item a, the underlying assumption is that we are clustering 5 elements into one group at each level, and correspondingly, the radii constraint is 1.5 in point b. If the fraction number is increased, then the radius constraint should be lowered. For most datasets that we analyzed, these empirically chosen parameters did not affect the results appreciably. The clustering procedure is done is a multi-scale manner, such that we compute clusters of different scales at different levels. As long as the initial parameters (for the first level) are chosen on a sufficiently small scale, the entire procedure scales up automatically.

4.3 Orthogonal Basis

Given that we are dealing with a discrete space, it is natural to quantify smoothness using a Haar-like wavelet basis in terms of point-wise exponential decay of coefficients. The Haar wavelet is a sequence of rescaled "square-shaped" functions that together form a wavelet basis. With this in mind, we induce a Haar-like point-wise wavelet basis using a hierarchical partition tree described by Coifman & Maggioni (2006). The row space of the data matrix is denoted V_r. For each level $l,$ let V_r^l denote the space of all functions that are constant over all nodes at this level. For example, V_r^1 is a one-dimensional space of constant functions on X_r. The following sequence of subspaces

$$V_r^1 \subset \cdots \subset V_r^l \subset V_r^{l+1} \subset \cdots V_r^L = V_r \tag{25}$$

provides a basis for multi-resolution analysis of V_r. Let W_r^l denote the orthogonal complement of V_r^l in V_r^{l+1} ($V_r^{l+1} = V_r^l \oplus W_r^l$, where the symbol "$\oplus$ " denotes the union of two orthogonal complement spaces). Using this notation, the space of all functions V_r can be decomposed as follows

$$V_r = V_r^L = [\oplus_{l=1}^{L-1} W_r^l] \oplus V_r^1. \tag{26}$$

Consider a node in the partition tree X_k^l at level l that is split into two sub-nodes, X_j^{l+1} and X_{j+1}^{l+1}, respectively. There is a zero-mean Haar-like function $\psi_{j,l,k}$ supported only over these two sub-nodes, and is piecewise constant on each of them. If a node X_k^l is split into three or more sub-nodes, then $(\#\rho(l,k) - 1)$ Haar-like orthogonal functions $\{\psi_{j,l,k}\}_{j=1}^{\#\rho(l,k)-1}$ must be constructed, where $\#\rho(l,k)$ denotes the cardinality (size) of the kth sub-node at level l. The collection of these functions is augmented by the constant function on X_r to form the following orthonormal basis of V_r

$$\Psi_r = \{\psi_0\} \cup \bigcup_{l=1}^{L-1} \bigcup_{k=1}^{K_l^r} \left[\{\psi_{j,l,k}\}_{j=1}^{\#\rho(l,k)-1}\right]. \tag{27}$$

In this equation, l denotes the level of the tree, k is the index of node X_k^l at level l, and $j = 1, \dots, \#sub(l,k) - 1$ (Gavish et al. 2010).

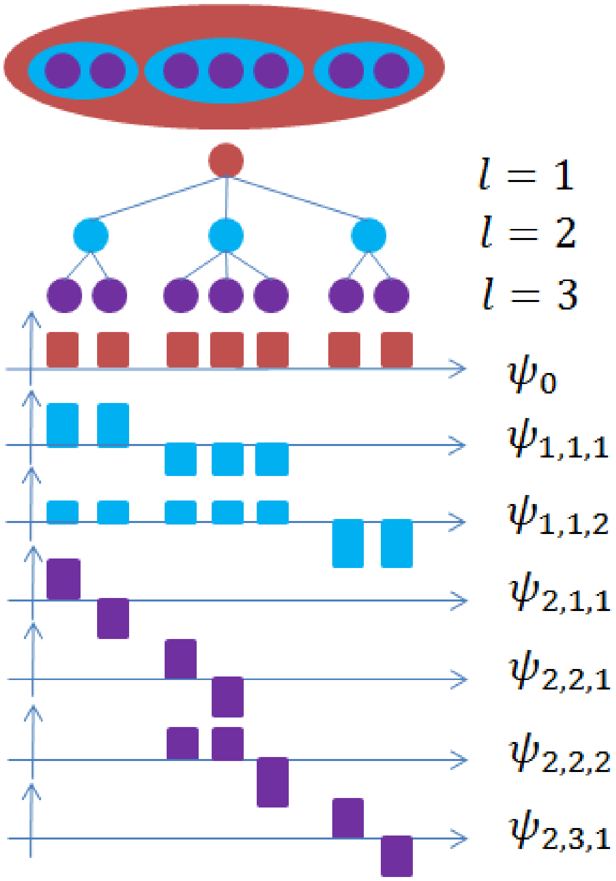


Figure 6. Hierarchical Partition Tree and Haar-like basis

Chapter 5 Dual Network and Co-clustering

Co-clustering is a mechanism of clustering a dataset in both the observation and the variable space. Specifically, given a data matrix, a co-lustering algorithm groups the rows of the matrix as well as the columns of the matrix into clusters. The critical key to solve the problem involving the selection of an efficient and accurate clustering algorithm, and a way of couple the row and column space.

In this chapter, we introduce the proposed co-clustering frame work and the supporting theory. First we introduce the schema to enhance the dual space. An organization in one space introduces stronger affinity between certain features in the other space, and vice versa. Secondly, we describe the co-clustering frame work. The diffusion embedding is applied before the clustering, so that the dimensionality is reduced and distance is accurate and reliable. The co-clustering algorithm is iterative with row clustering and column clustering conducted alternatively, and the dual enhancement appending step in between. We also give the theoretical support of how to stop the iterative process based on orthogonal basis expansion and harmonic analysis. Finally, we show a few examples of how to use the proposed method on both synthetic toy dataset and a real world dataset. A summary is given in the end of the chapter with conclusion and discussion of the proposed method.

5.1 Dual Enhancement

In the process of constructing the affinity graph, we treated each of the observation as a data point and computed the pairwise sparse affinity using the proposed affinity graph. Say if we use a cosine kernel with a zero threshold

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \max\left(\frac{< \boldsymbol{x}_i, \boldsymbol{x}_j >}{\|\boldsymbol{x}_i\|\| \boldsymbol{x}_j \|}, 0\right). \tag{28}$$

In a binary feature table where the entries are either “+1” or “-1”, the sparse affinity is simply the number of matching items minus the number of nonmatching items, normalized by the total number of variable. In this way, we are treating all the variables equally and even weighted, and independent. At the same time, we are assuming the observations are in an unorganized structure. But in reality, there is a way of organizing the observations such that it reveals a different weighting of the variables, further leading to an enhancement of affinities between parts of the variables. We introduced a scheme to enhance the dual affinity by introducing “meta” variables as a function of the organized observations’ variables.

A simple but persuasive example is given below in Table 1. In the table, we have four observations (let’s say four cells, C1, C2, C3, and C4), each with 3 variables F1, F2, and F3. The last row of the table contains the affinity of F1 to the each of the other variable. From the table, the information we derived is: The correlation between F1 and F2, and the correlation between F1 and F3 are equal. There is no preference or significant coherence from a global perspective.

If more information are given, say that we are informed of the clusters between C1 and C2, C3 and C4. By taking a function of the variables (we choose to use the average here), a table with additional coordinates are given as below in Table 2.

Table 1. Feature affinity without dual enhancement

	F 1	F 2	F 3
C 1	+1	-1	+1
C 2	+1	+1	+1
C 3	-1	+1	+1
C 4	+1	+1	-1
Affinity to F 1	1	0	0

Table 2. Feature affinity with dual enhancement

	F 1	F 2	F 3
C 1	+1	-1	+1
C 2	+1	+1	+1
C 3	-1	+1	+1
C 4	+1	+1	-1
{C 1, C 2}	+1	0	+1
{C 3, C 4}	0	+1	0
{ C1, C2, C3, C4}	+0.5	+0.5	+0.5
Affinity to F 1	1	1/21	5/21

In the new table with “meta” variables, the dimensionality of the feature is 7 instead of 4. The affinity computed suggests that there is a stronger affinity between F1 and F3 than between F1 and F2. The underlying interpretation is even the variable pairs have the same affinity in a small scale, but their affinity in a coarser is different.

Therefore, the organization in the observation space contains and provides affinity enhancement in the variable space in a coarser scale. In our method below, we use this additional coordination (“meta” variable) in clustering in dual space. The enhancement significance relies on the performance of the clustering result in the other space, and as well as the nature of the data.

5.2 Co-clustering Framework

Given the a data matrix $\boldsymbol{X}^{n\times m}$ with n rows and m columns, our goal is to cluster the rows and columns of the feature matrix concurrently, so that groups of similar objects and groups of correlated variables are identified at the same time. For this, we describe an iterative algorithm that clusters the rows and columns alternately. The core step of this algorithm is presented in “pseudo code” form as Algorithm 1 below, and described in detail next.

In the first step of this algorithm, each row of $\boldsymbol{X}$ is treated as a multivariate data point with dimension m. We assume that the structure of these data points can be modeled by a hierarchical partition tree denoted $T_r = \bigcup_{l=1}^{L} T_r^l$, where T_r^l is the lth level of the tree. The construction process of the partition tree is described in Chapter 4.

Once the row clustering is complete, an augmented data matrix, denoted X_r^A, of size $(n + \sum_{l=1}^{L-1} K_l^r) \times m$ is constructed as follows: X_r^A is the same as X_r, with newly appended rows being constituted of cluster centroids from all levels.

This augmentation has the effect of coupling the row and column spaces, allowing us to perform an analogous clustering of the columns. Specifically, we now treat each column of X_r^A as a data point with dimension $(n + \sum_{l=1}^{Lr-1} K_l^r)$. Next, denoting the column space of X_r^A as X_c, we construct a separate hierarchical partition tree T_c based on the column space X_c using the same procedure as for the rows. Similarly, an augmented dataset X_c^A is derived with a size of $n \times (m + \sum_{l=1}^{Lc-1} K_l^c)$ in the fourth step to set the stage for clustering the rows. By iteratively repeating the four steps listed above, the algorithm has been shown to converge to a stable and smoothly reorganized data matrix. In this matrix, clusters along the row space represent groups of similar cells, while clusters of columns represent groups of correlated features.

```
Algorithm 1: Co-clustering Iteration

Input: Data matrix $X^{n \times m}$

1. Denoting $X_r$ as the row space of $X$, construct a hierarchical partition tree
   $T_r$ on $X_r$.
2. Augment reordered dataset $X_r$ to form $X_r^A$.
3. Denoting $X_c$ as the column space of $X_r^A$, construct a hierarchical partition tree
   $T_c$ on $X_c$.
4. Augment reordered dataset $X_c$ to form $X_c^A$.
```

After the last iteration, an agglomerative hierarchical clustering is performed on each cluster at each level on both rows and columns, to reorder the data points. We use a simple average linkage based clustering algorithm, and the Euclidean distance metric during this last hierarchical clustering step (Sibson, 1973).

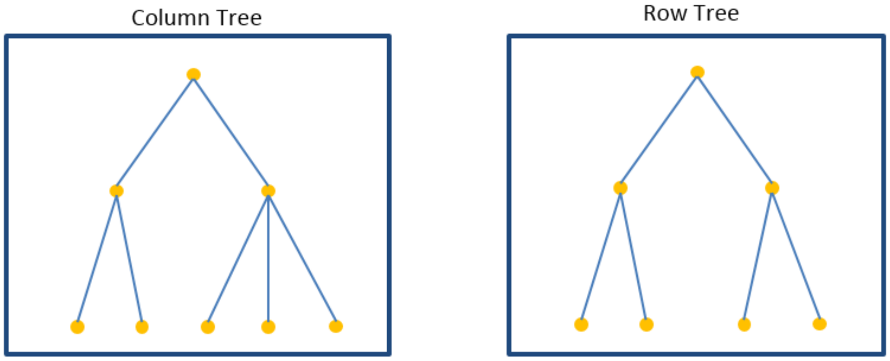


Figure 7. Row and column partition tree

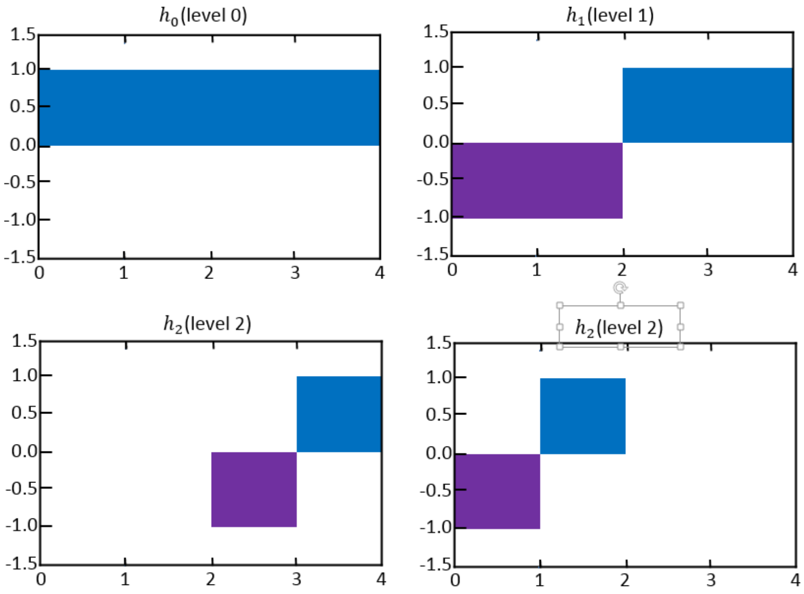


Figure 8. Row basis

5.3 2D Orthogonal Basis

In Chapter 4, Section 4.4, we described how to form a Haar-like orthogonal basis from a hierarchical partition tree. In the following paragraphs, we introduce the formation of 2-D Haar-like orthogonal basis derived from the 1-D basis in row partition tree and column partition tree. The 2-D basis are a set of tensor product of the 1-D basis. The tensor product of Ψ_c and Ψ_r (the column and row space basis functions, respectively), given by

$$\Psi = \left\{\prod \Psi_r \, \Psi_c\right\}, \tag{29}$$

where

$$\Psi_r = \{\psi_0\} \cup \cup_{l=1}^{L-1} \cup_{k=1}^{K_l^r} \left[\left\{\psi_{j,l,k}\right\}_{j=1}^{\#\rho(l,k)-1}\right], \tag{30}$$

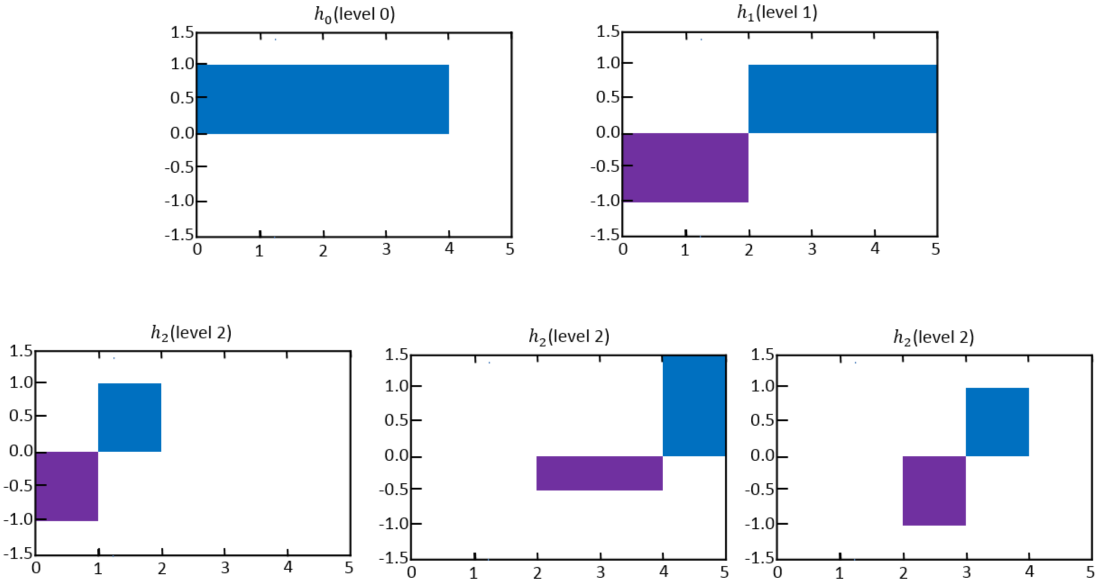


Figure 9. Column basis

and

$$\Psi_c = \{\psi_0\} \cup \bigcup_{l=1}^{L-1} \bigcup_{k=1}^{K_l^c} \left[\left\{\psi_{j,l,k}\right\}_{j=1}^{\#\rho(l,k)-1}\right]. \tag{31}$$

forms an orthogonal basis for the entire data matrix.

Assume a random row tree and column tree as shown in Figure 7, Ψ_r and $\Psi_c$derived from the 2 tree structures are shown in Figure 8 and Figure 9. The tensor product of the two 1-D basis then forms the 2-D orthogonal basis illustrated in Figure 10.

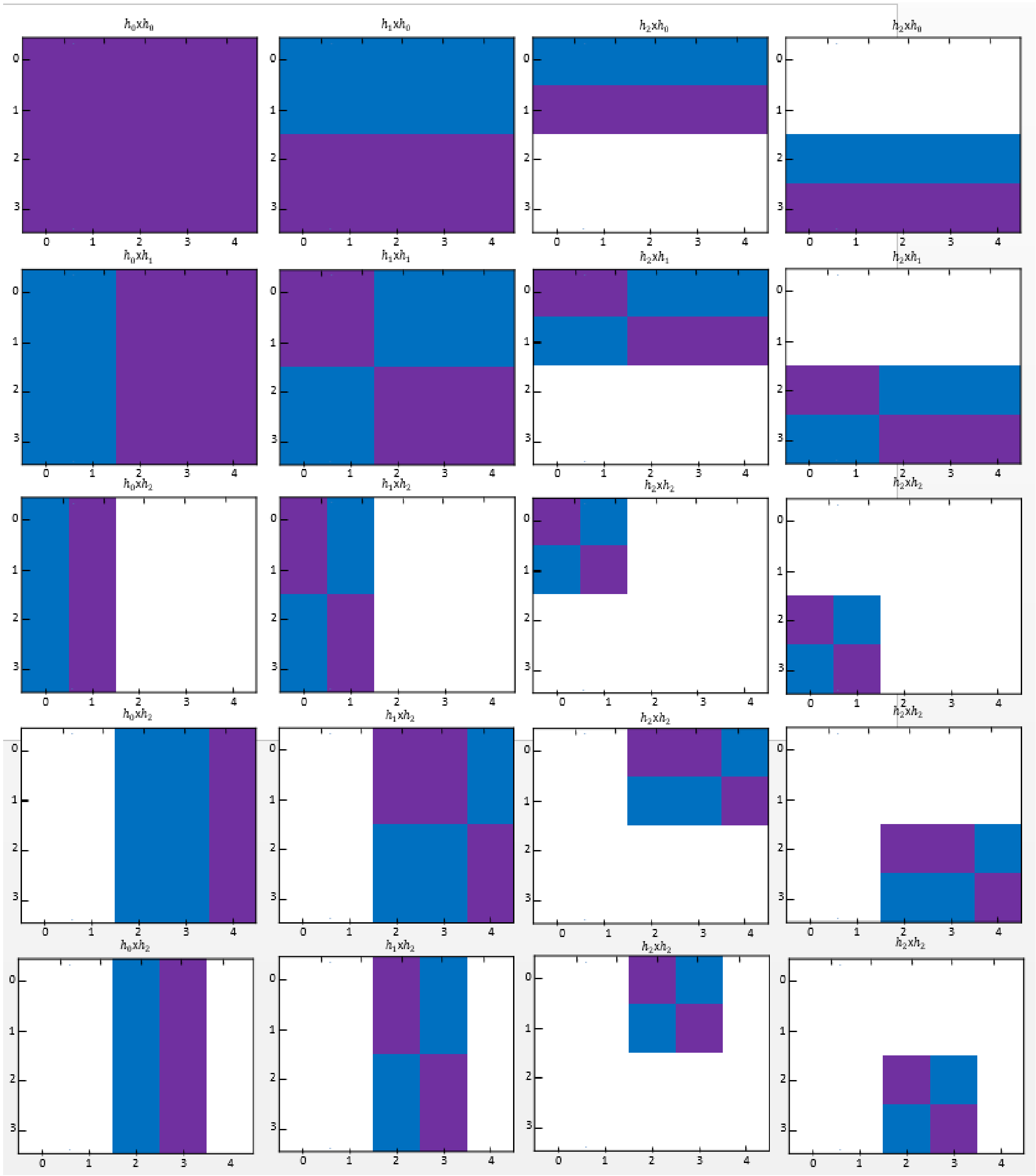


Figure 10. 2D rectangle basis

5.4 Co-clustering with Stop Criterion

This section describes the theory for sensing convergence of the iterative co-clustering procedure described above, so it can be halted correctly. Our goal is to arrive at a homogeneous set of clusters, despite noise in the data matrix. Achieving this goal requires a method for smoothing the data matrix. Therefore, we approximate the feature matrix by expanding it using this orthogonal basis after each iteration. The coefficients of this expansion, $\langle \boldsymbol{X}, \Psi_r \otimes \Psi_c \rangle$, decay rapidly and reach a plateau with increasing order of terms. It is known that the l_p sum of expansion coefficients, written as $\sum_{i,j} |\langle X, \Psi_r \otimes \Psi_c \rangle|^p$, indicating the smoothness, converges to a fixed value when further smoothing of the data matrix is not possible (Strömberg, 1998). In the special case when $p = 1$, the l_p sum is called the l_1 entropy. In our work, we use the l_1 entropy as the stopping criterion for the iterations. The iterative procedure using the l_1-entropy as a stopping criterion is summarized below as "Algorithm 2".

Algorithm 2: Harmonic based convergence

Input: Data matrix $X^{n \times m}$

1. Apply an iteration of Algorithm 1 and obtain partition trees T_r and T_{c} on row space and column space.
2. Construct Haar-like bases Ψ_{r} and Ψ_{c}.
3. Obtain tensor basis Ψ by taking the tensor product of Ψ_r and Ψ_c.
4. Compute the l_1-entropy $l_1(\Psi, X) = \sum |\langle X, \Psi \rangle|$.
5. Repeat steps 1 through 4 until $l_1(\Psi, X)$ converges.

5.5 Test Example

We evaluated the harmonic co-clustering method on synthetic datasets. The purpose of the synthetic data experiments is to confirm that the proposed co-clustering algorithm produces the correct results when the expected results are known in advance. The real-data experiments were designed to evaluate the practical usability of the proposed method on publicly available arbor reconstructions.

For the synthetic data, we evaluated the performance for one-way clustering and two-way co-clustering. For the neuronal datasets, we used the average F-measure of the (cross-validation) classification results to measure the clustering performance (Powers, D. M. W., 2011). We also computed a variety of other performance measures (described below) to measure the partitioning ability of our co-clustering method.

A synthetic block-structured data matrix with 200 rows and 200 columns was created for evaluating the correctness and performance of the presented algorithm. This matrix was created such that there are 3 known bi-clusters in the data. It consists of 3 major blocks, the sizes of which are $90 \times 70, 70 \times 80$, and 120×75, respectively. The first block consists of 90 data points drawn from 3 multivariate normal distributions with total dimensionality of 70. Of these 90 data points, 45 were drawn from $N(\boldsymbol{\mu}_{11}, \Sigma_{11})$. The next 30 data points were drawn from $N(\boldsymbol{\mu}_{12}, \Sigma_{12})$. And the remaining 15 points were drawn from $N(\boldsymbol{\mu}_{13}, \Sigma_{13})$. The second block consists of 80 data points from 2 multivariate normal distributions with dimensionality of 80. Half of them were drawn from $N(\boldsymbol{\mu}_{21}, \Sigma_{21})$ while the other half were drawn from $N(\boldsymbol{\mu}_{22}, \Sigma_{22})$. The third block consists of 120 data points from a multivariate normal distribution with dimensionality of 75, a sub block of it was replaced with another distribution of dimensionality 50. These parameters are summarized in Table 3.

Table 3. Synthetic data parameters

$\boldsymbol{\mu}_{11}$	$(8,8,\ldots,8)^T$	$\boldsymbol{\Sigma}_{11}$	$3*\boldsymbol{I}$
$\boldsymbol{\mu}_{12}$	$(9,9,\ldots,9)^T$	$\boldsymbol{\Sigma}_{12}$	$2*\boldsymbol{I}$
$\boldsymbol{\mu}_{13}$	$(10,10,\ldots,10)^T$	$\boldsymbol{\Sigma}_{13}$	$2.5*\boldsymbol{I}$
$\boldsymbol{\mu}_{21}$	$(4,4,\ldots,4)^T$	$\boldsymbol{\Sigma}_{21}$	$4*\boldsymbol{I}$
$\boldsymbol{\mu}_{22}$	$(2,2,\ldots,2)^T$	$\boldsymbol{\Sigma}_{22}$	$2*\boldsymbol{I}$
$\boldsymbol{\mu}_{31}$	$(-4, -4,\ldots,-4)^T$	$\boldsymbol{\Sigma}_{31}$	$3*\boldsymbol{I}$
$\boldsymbol{\mu}_{31}$	$(-6,-6,\ldots,-6)^T$	$\boldsymbol{\Sigma}_{31}$	$3*\boldsymbol{I}$

The rest of the matrix was set to background. Gaussian noise was added to the matrix to test the robustness of the algorithm. The synthetic matrix is displayed in Figure 11(A). This matrix was then permuted randomly. Figure 11(B) shows the permuted matrix in heat map form. This permuted matrix is presented as input to our co-clustering algorithm and the benchmark algorithm we compared with. One benchmark is a recent two-way hierarchical clustering algorithm (Chen et al., 2013), the results from which are shown in Figure 11(C). Average linkage and Euclidean distance was used in this algorithm to produce the displayed result. The algorithm was able to identify the first major block, but failed to identify the other two due to the presence of sub-clusters in each block. Figure 11(D) shows the results produced by our algorithm using diffusion distance and hierarchical reordering. The result is a perfect recovery of the generated data in Figure

11(A). It is also much smoother. The algorithm is able to identify the three major blocks correctly. One can easily find each of the sub-clusters and sub-blocks in the reorganized heat map, corresponding to the lower-level tree branches. Specifically, Figure 11 demonstrates the superiority of hierarchical harmonic co-clustering over Chen et al.'s two-way clustering algorithms: (A) Synthetic 200x200 data matrix displayed as a color-map. (B) Randomly permuted data matrix used as input to the co-clustering algorithm. (C) Chen et al.'s two-way hierarchical clustering fails to recover the data geometry. (D) Co-clustering results from the proposed algorithm are much more accurate.

We calculated the validity indices using the Fuzzy Clustering Toolbox (Balasko et al. 2005) for row clusters and column clusters to evaluate the one-way clustering performance in Figure 12. Here, SC (Partition Index) is the ratio of the sum of compactness and separation of the clusters. The Separation Index S uses a minimum distance separation for measuring partition validity (A. M. Bensaid et al., 1996). The XB (Xie & Beni, 1991) index quantifies the ratio of the total variation within clusters and the separation of clusters. The Dunn Index (DI) and Alternative Dunn Index (ADI) are for measuring compactness and cluster separation. Indices are calculated for clusters highlighted by the circles in Figure 11(C - D). Validity indices are lower for the proposed method indicating that it achieved a better clustering and partitioning result. Our method produces a superior co-clustering than the benchmark method (For both row and column cluster indices, four out five results are better).

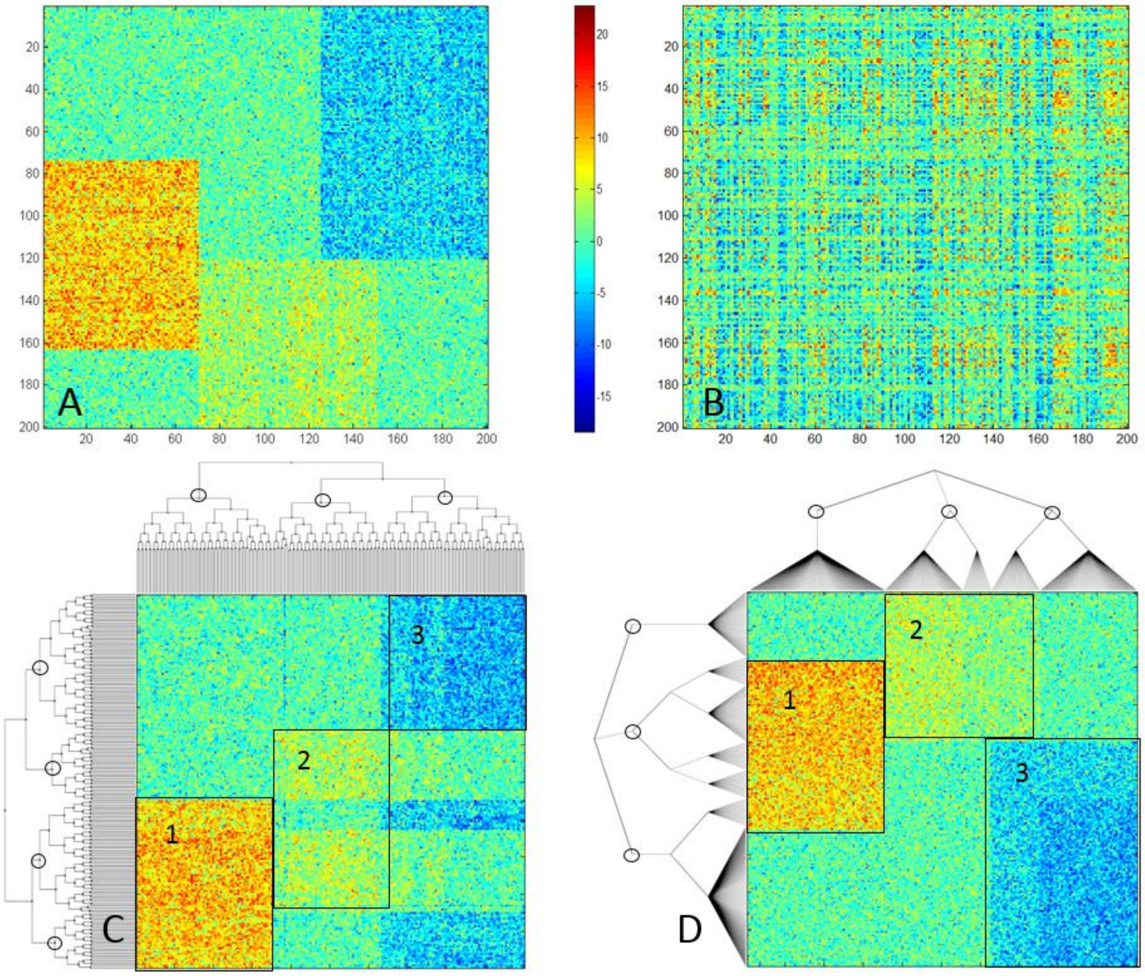


Figure 12. Demonstrating the superiority of hierarchical harmonic co-clustering: (A) Synthetic data matrix. (B) Randomly permuted data matrix. (C) Benchmark algorithm. (D) Results from the proposed algorithm are much more accurate.

We also use the Kullback-Leibler divergence (Kullback et al. 1951) to evaluate the two-way co-clustering performance for the dataset as shown in Table 4(E) (red bars correspond to the benchmark algorithm and the blue bars to our algorithm). The three chosen sub-blocks are indicated by the numbered rectangles. The diagonal entries with small K-L values in Table E show that the proposed algorithm's clusters are very homogeneous, whereas the non-diagonal elements with higher K-L values indicate that the groups are very dissimilar.

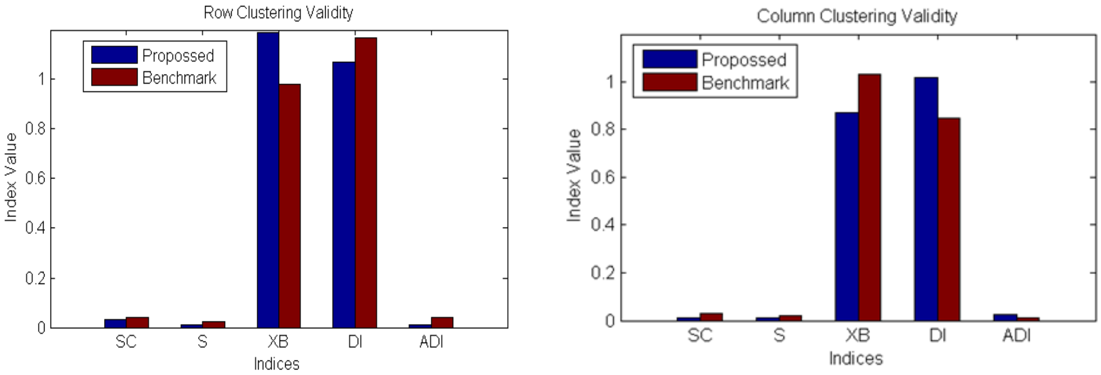


Figure 13. Row and column clustering validity indices SC: Partition Index, S: Separation Index, XB: Xie & Beni index, DI: Dunn Index, and ADI: Alternative Dunn Index indicate improved performance (smaller values are better).

Table 4. Kullback–Leibler (KL) divergence between group for Chen et al.'s algorithm in red and for the proposed algorithm in blue (smaller diagonal entries and higher off-diagonal entries indicate better performance).

	1	2	3
1	0.8/0.8	3.4/3.2	5.1/4.9
2	3.1/3.2	0.3/2.6	3.9/1.9
3	4.6/4.7	4.1/1.7	0.5/0.4

To further demonstrate the performance and the robustness of the proposed algorithm, we created a pool of synthetic datasets including 50 matrices with 200×200 entries each. All matrices are generated with different cluster patterns and background noise, but all the 50 synthetic datasets were created so they contain three row clusters and three column clusters, so the KL divergence numbers can be compared consistently in Figure 11 (A). For the compared method, we cut the dendrogram at the level that splits the data into three clusters. To evaluate the performance from all 50 datasets in a consistent manner, we define a validation score: KL divergence ratio, the ratio of the summed non-diagonal entries to the summed diagonal entries (Table 4). The distribution of the KL divergence ratios from the proposed algorithm and the benchmark algorithm as summarized by the histogram in Figure 12, clearly indicates the superiority of the proposed method.

5.6 Limitations of the Proposed Methods

The proposed co-clustering algorithm was able to recover the exact geometry structure in our synthetic data sets in the previous section, it also works well in identifying cell clusters and feature correlations in the experiments we analyzed on in Chapter 7. However, just like all algorithms in the literatures, our methods has its limits in certain conditions and data settings. Below we listed a few cases when our method might fail or the performance will not be ideal.

In a clustering problem, the algorithm performance will drop dramatically if there are more number of clusters then the number of data points in each cluster, the effect of the condition will be dramatic especially if the data points are not highly scattered across the feature space. Our method, especially in the process of construct the hierarchical partition tree, will suffer from the same reason.

Another case that our algorithm will not be necessary is when the data dimensionality is very low (typically less than 50 data points, and less than 10 in the feature dimensionality). The algorithm will still return a reasonable result, but it’s not necessary to apply such a comprehensive methods on the dataset with a simple setting.

In our methods, we have the ability to handle a reasonable amount of missing data, the typical way is to interpolating using the neighbor values. But when the amount of missing data increases and exceeds a certain level, the algorithm will lose its power in identifying the patterns and hidden geometries.

Chapter 6 Comparative Interpretation

In order to interpret the analysis and profiling result, in our framework, we propose to use the sparse group lasso as a feature selection and interpreting tool. In previous literatures, group lasso was first proposed by Mingyuan (2000). In their work, they considered the problem of selecting grouped variables (factors) for accurate prediction in regression. Such a problem arises naturally in many practical situations with the multifactor analysis-of-variance problem as the most important and well-known example. Instead of selecting factors by stepwise backward elimination, they focused on the accuracy of estimation and consider extensions of the lasso, the LARS algorithm and the non-negative garrotte for factor selection. Noah Simon presented a sparse group lasso algorithm in their work 2009. They discussed the sparsity and other regularization properties of the optimal fit for the model, and showed that it has the desired effect of group-wise and within group sparsity. They proposed the algorithm to fit the model via accelerated generalized gradient descent, and extended the model and algorithm to convex loss functions.

6.1 LASSO and Group LASSO

In 1996, Robert Tibshirani proposed a new method for estimation in linear models. The LASSO (Least Absolute Shrinkage and Selection Operator) is a regression method that involves penalizing the absolute size of the regression coefficients. The “LASSO” minimizes the residual sum of squares subject to the sum of the absolute value of the coefficients being less than a constant value. By penalizing (or equivalently constraining the sum of the absolute values of the estimates) you end up in a situation where some of the parameter estimates may be exactly zero. The larger the penalty applied, the further estimates are shrunk towards zero. This is convenient when we want some automatic feature/variable selection, or when dealing with highly correlated predictors. In our application, we choose to use lasso for selecting distinguishing measurement to interpret the profiling result and rebuild a model for further analysis.

In a typical linear regression model, assume the data of interest consists of N data points and each with p dimensions. The data matrix can be presented as $\boldsymbol{X} \in R^{N \times p}$, a respond (label) vector is usually given $\boldsymbol{y} \in R^{N}$. In many applications, we have $p \gg N$. To solve this, it is regularized by bounding the l_1 norm and minimize the objective function

$$\min_{\boldsymbol{\beta} \in R^{p}} (||\boldsymbol{y} - \boldsymbol{X\beta}||_2^2 + \lambda ||\boldsymbol{\beta}||_1). \tag{32}$$

In the equation above, $\boldsymbol{\beta}$ is the coefficients vector.

The solution of the above optimization problem yields a sparse coefficient vector $\boldsymbol{\beta}$, which in applications are interpreted as significant feature/measurement for the prediction/regression model.

Further, consider the application where features/measurements can be divided into practical meaningful groups, for example in a gene expression data, a group of genes are regulating the express of the same protein. Or a group of features are statistically correlated but independent among groups. If these group information is given or can be extracted from the dataset, a desired solution will be one that is able to give a sparse set.

Yuan & Lin proposed an algorithm for solving this problem. Suppose that the p predictors are divided into L groups, with p_l the size of each group. For ease of notation, we use a matrix $\boldsymbol{X}_l$ to represent the predictors corresponding to the lth group, with corresponding coefficient vector $\boldsymbol{\beta_l}$. Assume that $\boldsymbol{y}$ and X are normalized with zero mean. The algorithm is formulated as

$$\min_{\boldsymbol{\beta}\in R^p}(\left\|\boldsymbol{y}-\sum_{l=1}^{L}\boldsymbol{X}_l\boldsymbol{\beta}_l\right\|_2^2+\lambda\sum_{l=1}^{L}\sqrt{p_l}\|\boldsymbol{\beta}_l\|_2), \tag{33}$$

where the $\sqrt{p_l}$ terms accounts for the varying group sizes, and $\|.\|_2$ is the Euclidean norm (not squared). This procedure acts like the lasso at the group level: depending on λ, an entire group of predictors may drop out of the model. In fact if the group sizes are all one, it reduces to the lasso.

6.2 Sparse Group LASSO

In the above algorithm, however, we cannot achieve sparsity within groups. Specifically, if the coefficient for one group does not shrink to zero, all the member variables within the group will be non-zero. In our application, we need a regularization that yields sparsity both in group levels and individual variable levels as well. To achieve this, we adopt a sparse group lasso model proposed by (Noah), which provides for a more general criterion also works for the standard group lasso with non-orthonormal model matrices. Consider the sparse group lasso criterion

$$\min_{\boldsymbol{\beta}\in R^p}\left(\left\|\boldsymbol{y}-\sum_{l=1}^{L}\boldsymbol{X}_l\boldsymbol{\beta}_l\right\|_2^2+\lambda_1\sum_{l=1}^{L}\sqrt{p_l}\|\boldsymbol{\beta}_l\|_2+\lambda_2\|\boldsymbol{\beta}\|_1\right), \tag{34}$$

where = $\boldsymbol{\beta}=(\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,\boldsymbol{\beta}_3,\dots,\boldsymbol{\beta}_L\ )$is the entire parameter vector.

6.3 Criterion and Algorithm

The problem is formulated as a convex optimization problem. Therefore, the optimal solution is characterized by the subgradient equations. The criterion is separable so that block coordinate descent can be used for the optimization. Focusing on one group k, the subgradient must satisfy

$$\boldsymbol{X}_k{}^T\left(\boldsymbol{y}-\sum_{l=1}^{L}\boldsymbol{X}_l\boldsymbol{\beta}_l\right)=\lambda_1 s_k+\lambda_2 t_k, \tag{35}$$

where s_k and t_k are the subgradient of $\|\boldsymbol{\beta}_k\|_2$ and $\|\boldsymbol{\beta}\|_1$. $s_k=\frac{\boldsymbol{\beta}_k}{\|\boldsymbol{\beta}_k\|_2}$, and t_k = sign($\boldsymbol{\beta}_k$). It is obvious that when

$$\left\|\boldsymbol{X}_k{}^T(\boldsymbol{y}-\sum_{l\neq k}\boldsymbol{X}_l\boldsymbol{\beta}_l)-\lambda_2\right\|_2\leq\lambda_1,\ \boldsymbol{\beta}_k=0. \tag{36}$$

The subgradient equation can also give insight into the sparsity within a group which is at least partially nonzero. We redefine the variables as $\boldsymbol{X}_l=\boldsymbol{Z}=(Z_1,Z_2,\dots,Z_{p_l})$, and the coefficients $\beta_l=\theta=(\theta_1,\theta_2,\dots,\theta_{p_l})$. That is:

$$Z_j{}^T\left(\boldsymbol{y}-\sum_{l=1}^{L}\boldsymbol{X}_l\boldsymbol{\beta}_l\right)=\lambda_1\frac{\theta_j}{\|\boldsymbol{\beta}_k\|_2}+\lambda_2 t_j. \tag{37}$$

To satisfy $\theta_j=0$, only if

$$\left\|Z_j{}^T(\boldsymbol{y}-\sum_{l\neq k}\boldsymbol{X}_l\boldsymbol{\beta}_l-\sum_{i\neq k}Z_j\theta_j)-\lambda_2\right\|_2\leq\lambda_1. \tag{38}$$

Overall, the algorithm is a sequence of nested loops with outer loop solving group sparsity and inner loop solving within group sparsity:

Cyclically iterate through the groups, at each group (k) execute step (2)

Check if the groups' coefficients are identically 0, by seeing if they obey

$$\left\| \boldsymbol{X}_k{}^T(\boldsymbol{y} - \sum_{l \neq k} \boldsymbol{X}_l \boldsymbol{\beta}_l) - \lambda_2 \right\|_2 \leq \lambda_1$$

If not, within the group apply step (3).

Start with $\theta = \hat{\theta}$, check if $\left\| Z_j{}^T\left(\boldsymbol{y} - \sum_{l \neq k} \boldsymbol{X}_l \boldsymbol{\beta}_l - \sum_{i \neq k} Z_j \theta_j\right) - \lambda_2 \right\|_2 \leq \lambda_1$, if so, set

Chapter 7 Applications on Profiling Brain Cells

7.1 L-measure

As noted above, the L-measure (LM) (Scorcioni et al., 2008) is a set of neuroanatomical features (summarized graphically in Figure 13) that collectively characterize the morphology of an individual cell's arbor. The features are computed from arbor reconstruction files in the standard SWC file format. The L-measure extracts a set of approximately 40 core morphological measurements of neurons (e.g., soma size, number of stems, process diameters, lengths, bifurcation patterns), and their derived statistics (average, standard deviation, minimum, maximum, and sum). In this work, we extended the list of L-measure parameters to capture additional morphological characteristics of cells. In particular, we added features describing the wrapping convex hull and an ellipsoid that captures the overall shape and orientation of the cells, to derive a list of 130 features per cell in http://farsight-toolkit.org/wiki/L_Measure_functions. Figure 13 shows a graphical summary of the quantitative arbor measurements used for this study. The entries in green are derived from the L-measure of Scorcioni et al., (2008), and the entries in red are our extensions to the L-measure. Overall, there are 43 fundamental measurements, and additional derived measurements (e.g., mean, variance, maximum, minimum, etc.) that together amount to 130 measurements per reconstruction.

The multivariate input data to the co-clustering algorithm are denoted $X = \{\boldsymbol{x}_i\}_{i=1}^{n}$, where n is the number of data points (cells), and each vector $\boldsymbol{x}_i$ has as many elements as there are L-measure features (~130). These input data can also be written as a large feature matrix $X^{n\times m}$, with n rows corresponding to n cells, and m columns corresponding to m arbor features.

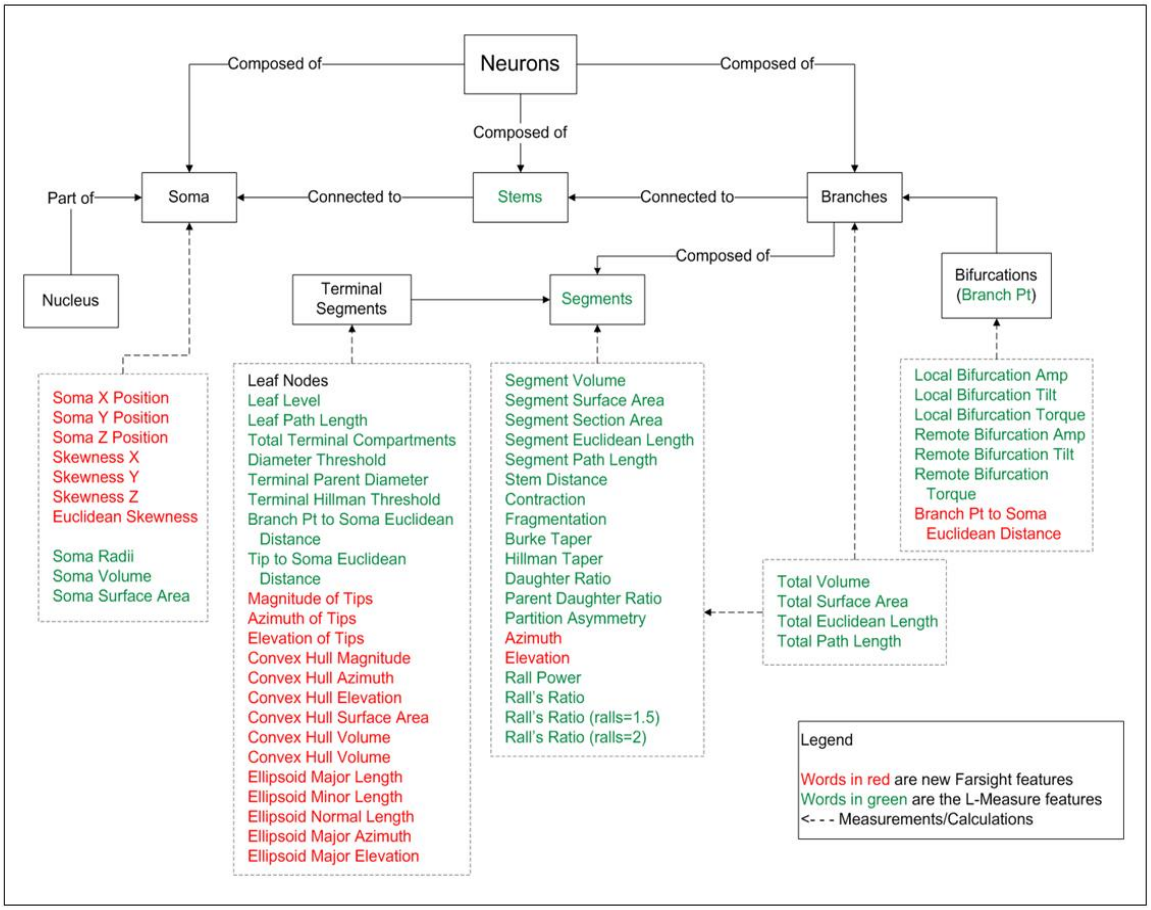


Figure 13. A graphical summary of the quantitative arbor measurements used for this study. The entries in green are derived from the L-measure of Scorcioni et al. (2008), and the entries in red are our extensions to the L-measure.

7.2 Mouse Brain Astrocytes Analysis

Glial cells account for a large fraction of the cells in the mammalian brain, with astrocytes constituting the most abundant (>50%) cell type (Verkhratsky & Butt 2007). Astrocytes are critical to brain development, physiology, and pathology, including: (i) regulation of neuro, glio- and synaptogenesis (Song et al., 2002), (ii) development and regulation of the blood-brain barrier, (iii) responding to various kinds of brain insults through reactive astrogliosis, (iv) mediating several neuronal and other diseases, such as HIV, depression, brain ischemia and edema, epilepsy, and dementia, and (v) reacting to foreign objects including neural probes. In executing these functions, astrocytes undergo alterations in structure, functional state, and relationship with other cells, including astrocytes, microglia, neurons, and endothelial cells. There has been a growing interest in understanding the roles of astroglia (Kulkarni, et al., 2015).

In our work, we primarily report results from the data acquired in a previously published study (Maynard & Leasure, 2013). For these examples, three 50 μm thick serial sections from the medial prefrontal cortex of 24 rats exposed to binge alcohol consumption and voluntary exercise were multiplex labeled to reveal cell nuclei (DAPI), microglia (Iba-1), vessels (SMI-71), neurons (NeuN), and astrocytes (GFAP). Eight confocal image tiles, forming a 2 × 4 montage covering a 775 µm × 1,550 µm area, were imaged for each tissue section, resulting in 24 tiles for each animal. The tiles were imaged sequentially in five fluorescence channels using the 405 nm, 488 nm, 560 nm, 594 nm, and 633 nm laser lines of a Leica SP8 upright confocal microscope with a 40× oil immersion objective. The image dimensions were 1,024 × 1,024 × 52 voxels (387.50 µm × 387.50 µm × 1 µm). The acquisition speed was 600 Hz, and the zoom factor was 0.75. The tiles were set to have approximately 10% spatial overlap, and the z-stacks were collected encompassing the entire thickness of the tissue.

Although the vital roles of astrocytes in brain development, physiology and pathology are receiving growing recognition, much remains unknown about their quantitative architecture, especially the heterogeneity in astrocyte arbor morphology. In order to overcome these barriers, it is essential to first quantify the arbor morphology. Unfortunately, no single number is capable of describing a complex three-dimensional cell arbor. For this reason, we adopt the L-measure introduced in the previous section.

With the large number of cells that are present in brain tissue, and the high-dimensionality of the L-measure's feature space, the analysis of the combined data from all of the cells is a non-trivial task. We applied the proposed method to the astrocyte arbor reconstructions obtained from the 24 datasets used in this study. Figure 14 demonstrates the co-clustering results in the form of a heat-map representation where each row corresponds to a cell, with a total of about 30,000 cells in this study, and each column corresponds to a feature from the L-measure collection. The horizontal tree on the left side of the heat-map illustrates the grouping structure of the cell population, whereas the tree on top illustrates the grouping structure of feature correlations. Figure 14 shows a quantitative analysis of GFAP+ cells from a binge alcohol study with 24 animals, four groups (24 fields, 387.07 × 387.07 × 50 μm were imaged from each animal), using co-clustering of L-measure data. (A) Heat map rendering of the co-clustering result (each row corresponds to an individual cell, each column represents a feature). Four clusters of morphological cell-types were identified (circled in the horizontal tree) in (G1 – G4). Representative cells from each group. This figure indicates that the population under study consists of cells which have a wide variability in terms of arbor morphology, starting with complex morphology with multiple branches in G1 to relatively simple structures in G4.

The co-clustering reveals that the GFAP+ cells in the population under study fall into four major groups, highlighted by circles in the horizontal tree in Figure 14. Representative cells from these four groups are shown in columns (G1 - G4). Results

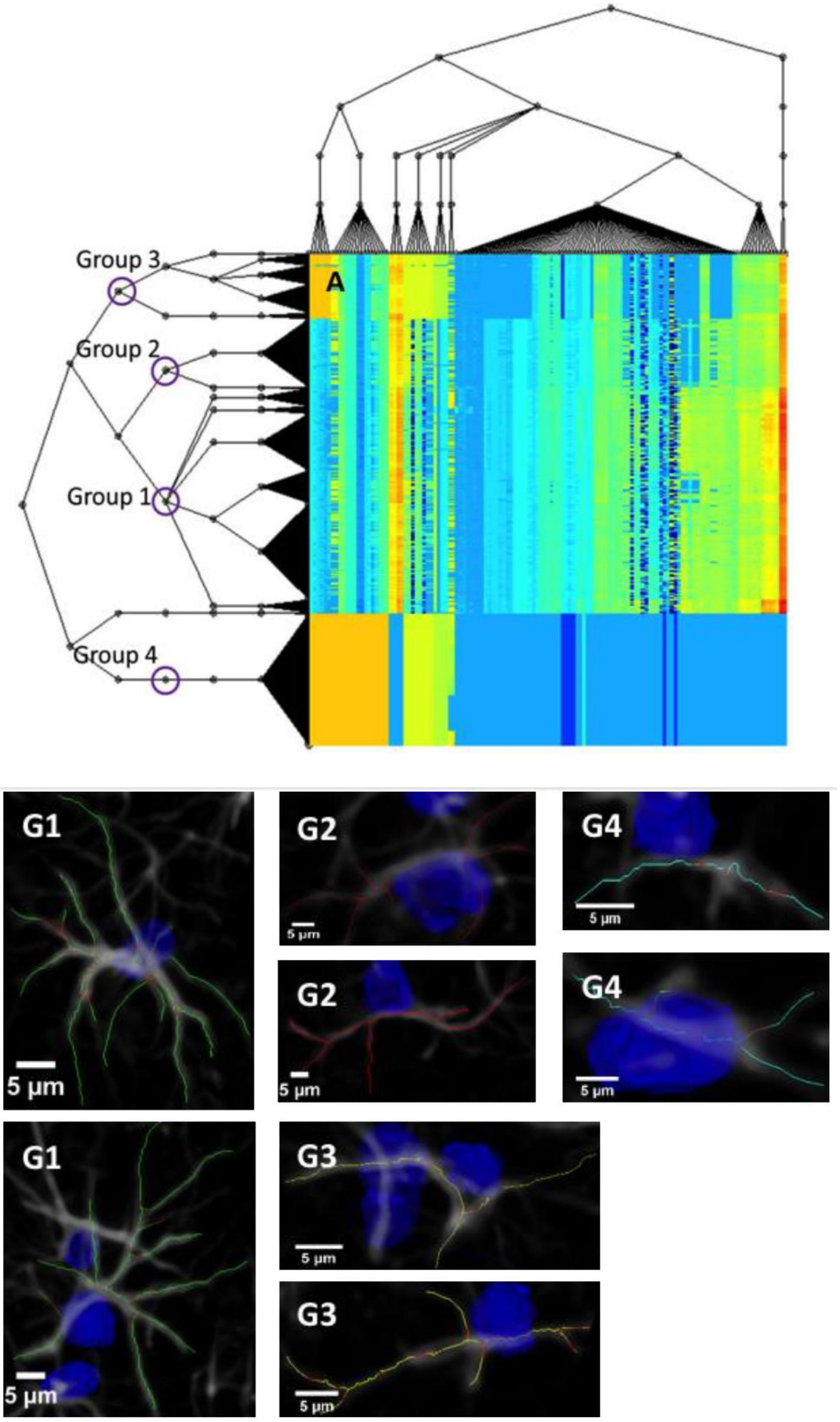
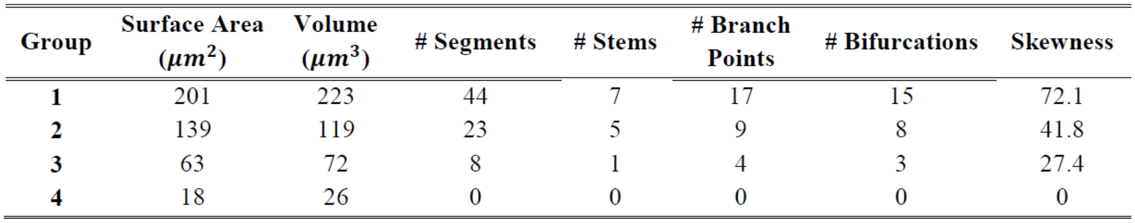


Figure 15. Quantitative analysis of a binge alcohol study with 24 animals, four groups, using co-clustering of L-measure data. (A) Heat map rendering of the co-clustering result (each row corresponds to an individual cell, each column represents a feature).

Table 5. Features selected by the co-clustering algorithm. Results indicate that the astrocyte cells in the given population vary largely in terms of their size, shape and arbor complexity.



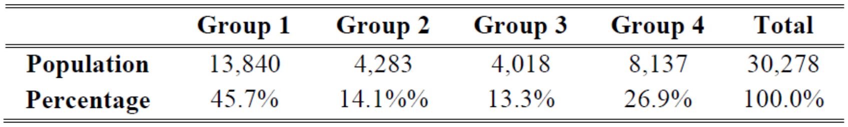
Group	Surface Area (μm^2)	Volume (μm^3)	# Segments	# Stems	# Branch Points	# Bifurcations	Skewness
1	201	223	44	7	17	15	72.1
2	139	119	23	5	9	8	41.8
3	63	72	8	1	4	3	27.4
4	18	26	0	0	0	0	0

indicate that the four groups represent cells with variable morphological complexity, with the most complex arbors in G1 to the least complex in G4.

Table 5 lists the mean values of the relevant features for each group. It can be observed that the four groups are distinct in terms of the overall sizes of the cells (as indicated by the surface area and volume), the complexity of their arbor morphologies (as indicated by the number of segments, stems, branch points, and bifurcations) and their shapes (as indicated by the skewness values).

Table 6 lists the proportions of the four groups within the population under study. It can be observed that the GFAP+ astrocyte cell population consists largely of cells with high complexity G1 and low complexity G4.

Table 6. Astrocyte cell distribution by group. The population under study is seen to consist mostly of cells with complex arbor morphologies (Group 1).



	Group 1	Group 2	Group 3	Group 4	Total
Population	13,840	4,283	4,018	8,137	30,278
Percentage	45.7%	14.1%%	13.3%	26.9%	100.0%

In order to quantify the performance of co-clustering, a set of statistical dispersion indices are computed, by the co-clustering algorithm based on the diffusion distance. The within-cluster dispersion index is defined as

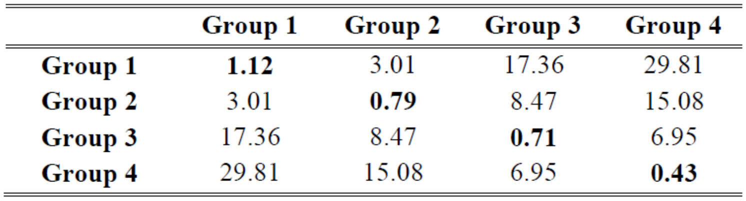
$$p_{intra,k} = \frac{\sum_{i=1}^{N_k}(d_{ik} - r_k)^2}{r_k}, \tag{39}$$

where $pintra,k$ is the intra cluster dispersion of cluster k, rk is mean of distances between all data points and the centroid of cluster k, and dik is the distance between data point i and the centroid of cluster k. Similarly, the inter-cluster dispersion index is defined as

$$p_{inter,pq} = \frac{\sum_{i=1}^{N_q}(d_{iq} - r_{pq})^2 + \sum_{i=1}^{N_p}(d_{ip} - r_{pq})^2}{r_p + r_q}, \tag{40}$$

where $pinter,pq$ is the inter cluster dispersion of cluster p and q, rpq is the mean distance between all data points in cluster p and the centroid of cluster q and rqp is the mean distance between all data points in cluster q and the centroid of cluster p. Table 7 shows the intra and inter-cluster dispersion indices computed for the four groups. The intra-cluster indices with low values indicate strong intra-cluster homogeneity, while high values of the inter-cluster dispersion indicates strong cross-cluster heterogeneity.

Table 7. Inter and intra-cluster dispersion values for the four cell groups. Low intra-cluster dispersion indicates strong within-cluster homogeneity while high inter-cluster dispersion indicates strong between cluster heterogeneity.



	Group 1	Group 2	Group 3	Group 4
Group 1	**1.12**	3.01	17.36	29.81
Group 2	3.01	**0.79**	8.47	15.08
Group 3	17.36	8.47	**0.71**	6.95
Group 4	29.81	15.08	6.95	**0.43**

Finally, we note that unlike microglia, the morphological analysis of astrocytes is an emerging area of research (Zhang & Barres, 2010; Anderson et al., 2014) and new morphological categories are still being discovered (Matyash & Kettenmann 2010). Consequently, the interpretation of cell types is largely dependent upon the experimental hypothesis being tested. To this end, the proposed co-clustering method provides a powerful quantitative tool for investigating morphological heterogeneity of astrocyte cell populations.

7.3 Mouse Brain Microglia Analysis

The brains of Long Evans rats were dissected free of the skull while not disturbing implanted neural recording devices (NeuroNexus Inc., Ann Arbor, MI) after 30 days of recording. The brains were fixed, aligned and blocked to optimize the capture of each device within a single 100μm thick tissue slice. Histochemistry was used to label all cell nuclei (Hoechst 33342), astrocytes (GFAP), microglia (Iba-1), and neurons (NeuroTrace 640/660). The tissue slices were imaged using an Olympus IX81 inverted DSU microscope with a 30x silicone oil objective with 800μm working distance. Unimplanted control tissue was collected using the same protocols for comparison.

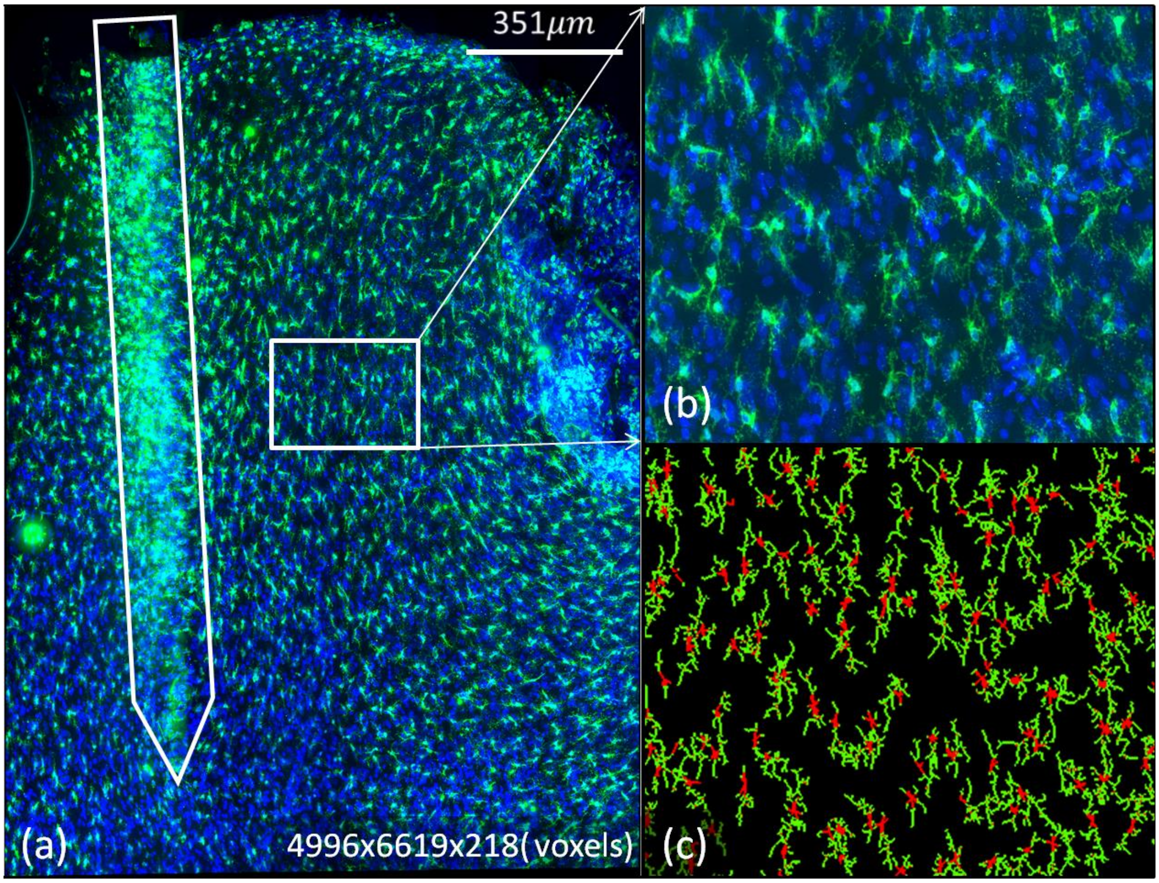


Figure 15. Fluorescent image and traces: (a) projection of 3D fluorescent image from device implanted tissue, the white arrow indicates the device region, (b) a patch from image a, (c) reconstructed traces.

Microglia consist of a central soma containing the cell nucleus, from which the cell arbor emanates. To segment the soma, we segment all cell nuclei, and isolate the nuclei that are positive for Iba-1. A level set-based segmentation is used to segment the somas for the isolated cells. An automated minimal spanning tree based algorithm is used to trace the cell arbors in Figure 15(c), after which their morphology is quantified using the L-measure library from computational neuroanatomy. A total of 127 features are computed for each cell arbor. Figure 15 shows a sample image, and the automated microglia traces. Two L-measure tables were thus generated, one from a device-implanted tissue (4,408 cells), and the other control tissue (3,891 cells).

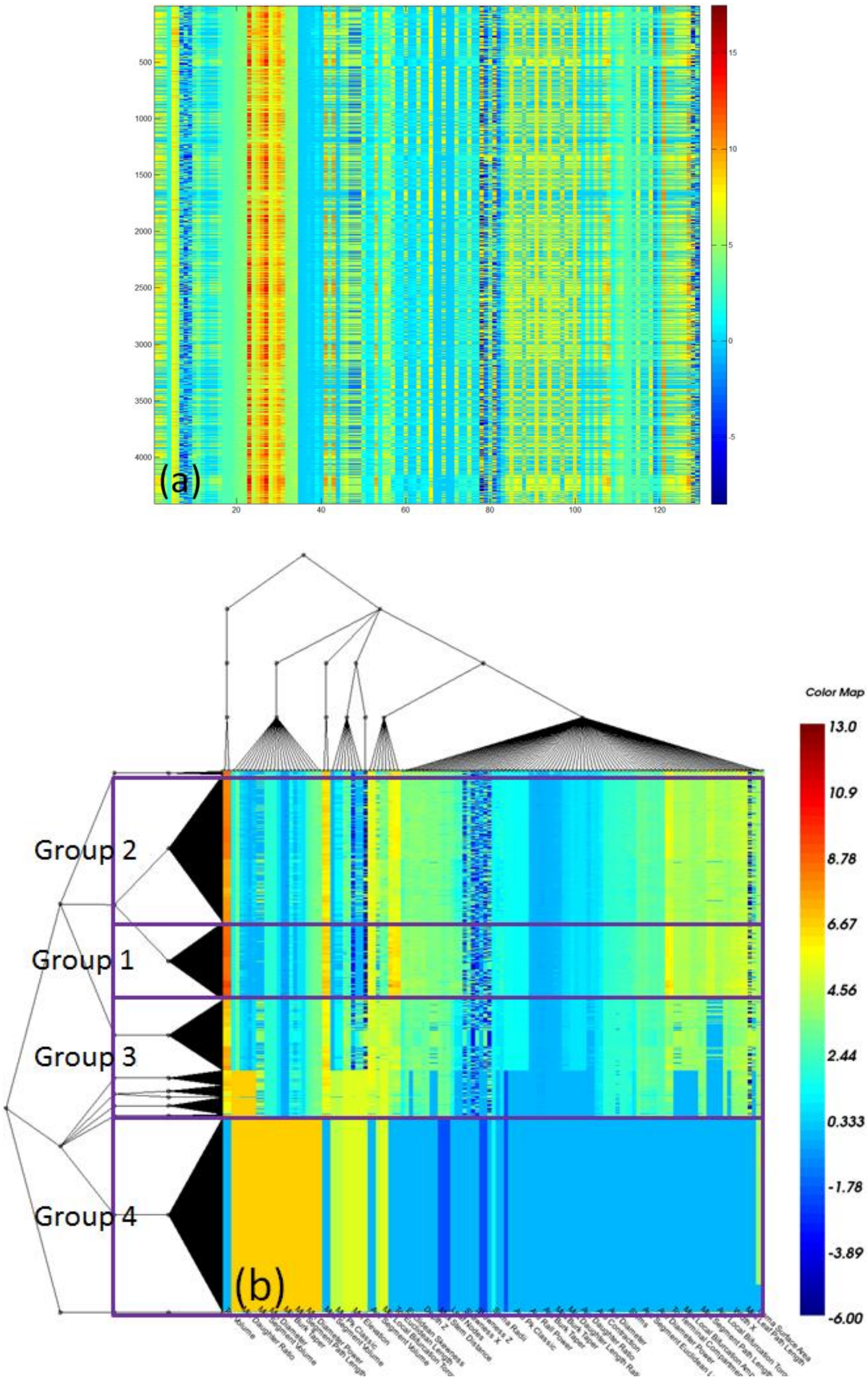


Figure 16. Heatmap of (a) raw feature table of the implanted tissue data, (b) reorganized data table applying Algorithm 1 Purple boxes indicated four salient groups of cells.

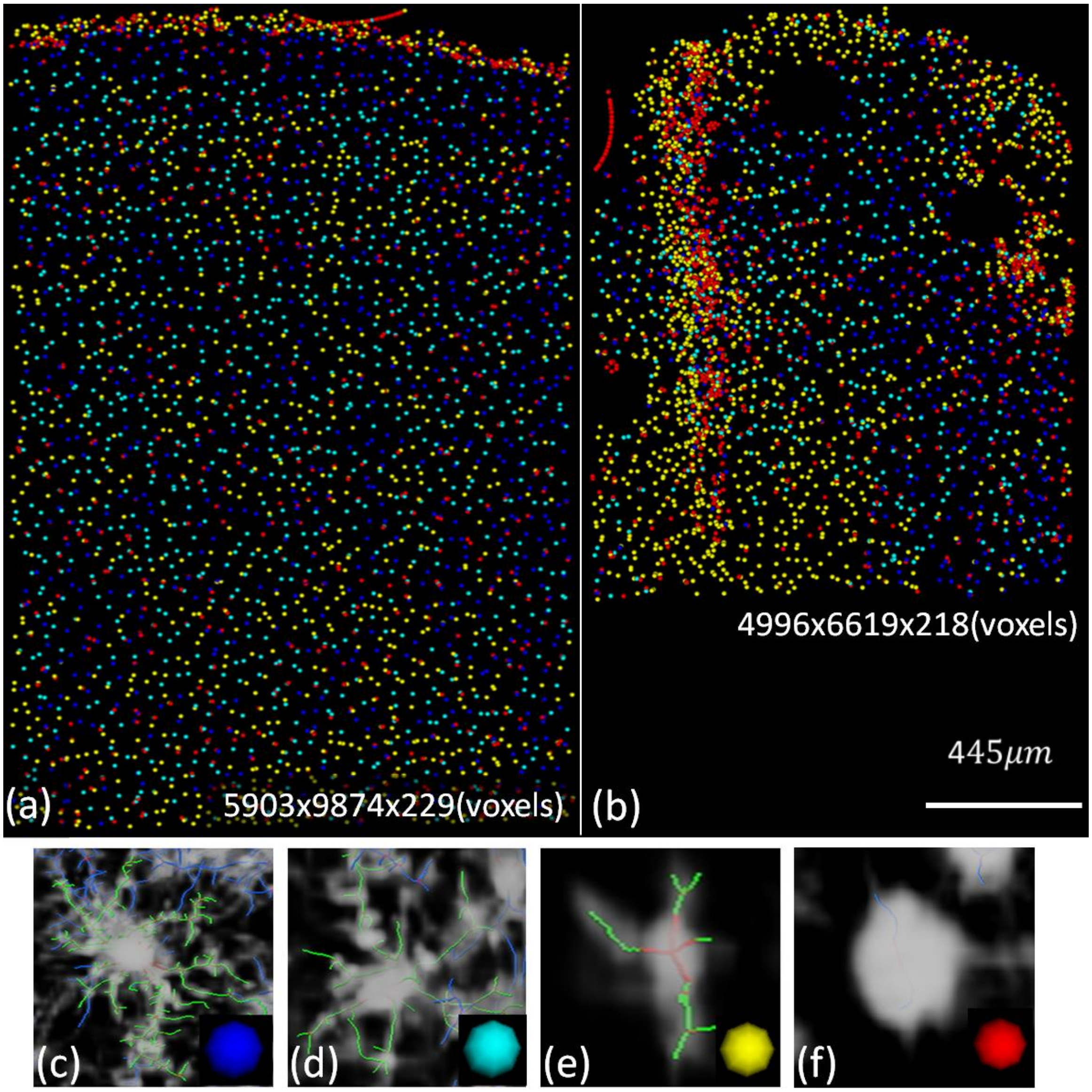


Figure 17. Spatial distribution & typical samples: (a-b) spatial distribution of four subsets of cells for a control tissue and implanted tissue, color coded spheres corresponding to cells in four groups, (c-f) typical samples of four subsets cells.

Figure 16(a) shows a heatmap representing the L-measure data matrix from the implanted tissue dataset, and the heatmap in Figure 16(b) shows the distribution of the results of applying Algorithm 1 to the data matrix, after convergence. In the heatmap, each row corresponds to a cell and each column corresponds to a feature. Each entry is mapped to a color according to the accompanying color map. The hierarchical partition trees on the row and column spaces are plotted adjacent the heatmap. In our software implementation, each row in the heatmap, and each node in the row partition tree are linked to the image visualization system, allowing each cell group to be highlighted for visualization.

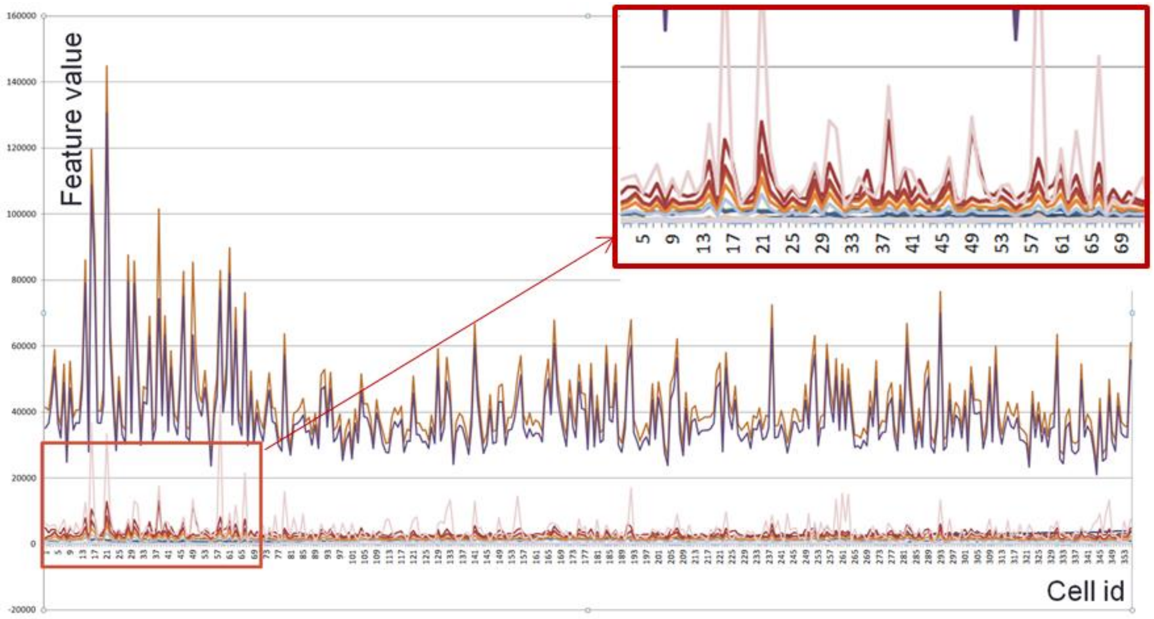


Figure 18. Correlated features: two groups of features plotted with high correlation within cluster.

The harmonic co-clustering reveals four salient groups of cells, as seen in Figure 17(b). Figure 17(a) – (b) show the spatial distributions of each of these cell groups using color-coded spheres for control, and device-implanted tissue, respectively. The panels (c) – (f) in Figure 17 show close-ups of sample cells from each cluster. Microglia exhibit varying arbor morphologies indicative of their activation states. Resting microglia have symmetric and complex arbors. Progressively more activated microglia exhibit less complex arbor morphologies. Clearly, in the control tissue, cells from the four cell groups are distributed uniformly throughout the tissue. However, in the device-implanted tissue, a group of complex (resting) cells is distributed far away from the device-implantation region, and the group of low-complexity (activated) cells and arborless round cells are close to the device-implantation region. The most salient features that distinguish each cluster from the others are arbor complexity and size features, and sample values are listed in Table 8. In this table, decreasing mean values of features along groups is visually consistent with the cell morphologies as seen in Figure 17.

Features are highly correlated with each other within each feature group. In Figure 20, the two groups of features are plotted along cells from one cluster.

We also analyzed a large dataset of nearly 30,000 cells that combines datasets from ten different tissues, four of them are control tissues with 14,570 cells and six are implanted tissues with 14,983 cells. The tissue perturbation caused by the device implantation can be quantified, as follows. The percentages of each subset of cells from ten datasets are shown in Table 9. The proportion of the first two groups (high and moderate complexity cells considered as resting cells) from implanted tissues is lower than that from control tissue, but the proportion of the last two groups (low complexity and round cells considered as activate cells and arbor less amoeboid) from implanted tissue is higher than that from control tissue. Thus the inserted device triggers microglia activation near the device.

Table 8. Variation of the Significant Features among Groups

Group	Features						
	Surface_Area	Volume	Segments	Stems	Branch_Point	Bifurcation	Skewness
1	7,433	8,226	34	7	14	13	29.8
2	3,478	3,770	17	5	6	5	26.9
3	1,773	1,928	10	3	5	3	17.3
4	50.3	33.5	0	0	0	0	0

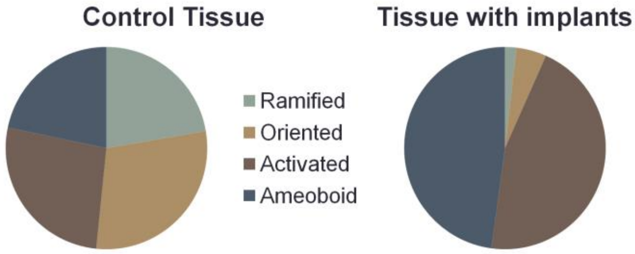


Figure 19. Identified group population in control and experimental tissue.

Table 9. Cell Proportion from Ten Datasets

	G 1	G 2	G 3	G4	Percentage Ratio
Device	274	725	6,815	7,169	1.9:4.9:45.4:47.8
Control	3,250	4,272	3,883	3,165	22.4:29.3:26.6:21.7

7.4 Neuronal Reconstruction Analysis

7.4.1 Pyramidal Neurons from the Mouse Neocortex

An ensemble of 728 reconstructions (all available entries) of six morphological subtypes of pyramidal neurons with wide-ranging morphologies were downloaded from the NeuroMorpho.Org (Ascoli et al., 2007) database (www.neuromorpho.org) to test the performance of the proposed method. All the reconstructions are in the open SWC format (Cannon et al., 1998). These reconstructions were from diverse sources including (Chen et al., 2009; Krieger et al., 2007; Rocher et al., 2010; Smit-Rigter et al., 2012; Lee et al., 2011; and Trevelyan et al., 2006). This ensemble is displayed in Figure 20.

The raw L-measure data for this ensemble is presented visually as a heat map in Figure. 21(A). In Figure 21(B), the reorganized data matrix resulting from the co-clustering algorithm is presented as a heat map. The horizontal tree structure shows the geometry of the dataset in terms of cells whereas the vertical tree structure indicates the geometry of the data set in terms of features. The six subtypes of cells are identified at the third level in the heat map. The corresponding cell groups are shown in Figure 21(D) and Figure 21(A).

For the analysis, the exact number of cell clusters and the number of feature groups are not required from the user. We set the initial values as 1/5 of the cell number and 1/5 of feature number for rows and columns respectively, these numbers could be set differently for other applications. Since clustering is conducted at multiple scales, groups obtained at different levels are at different similarity scales. The visualization (the heat map) and the linked system clearly identified valid groups. The algorithm is able to correctly identify the six subtypes of cells based on the morphological measurements alone.

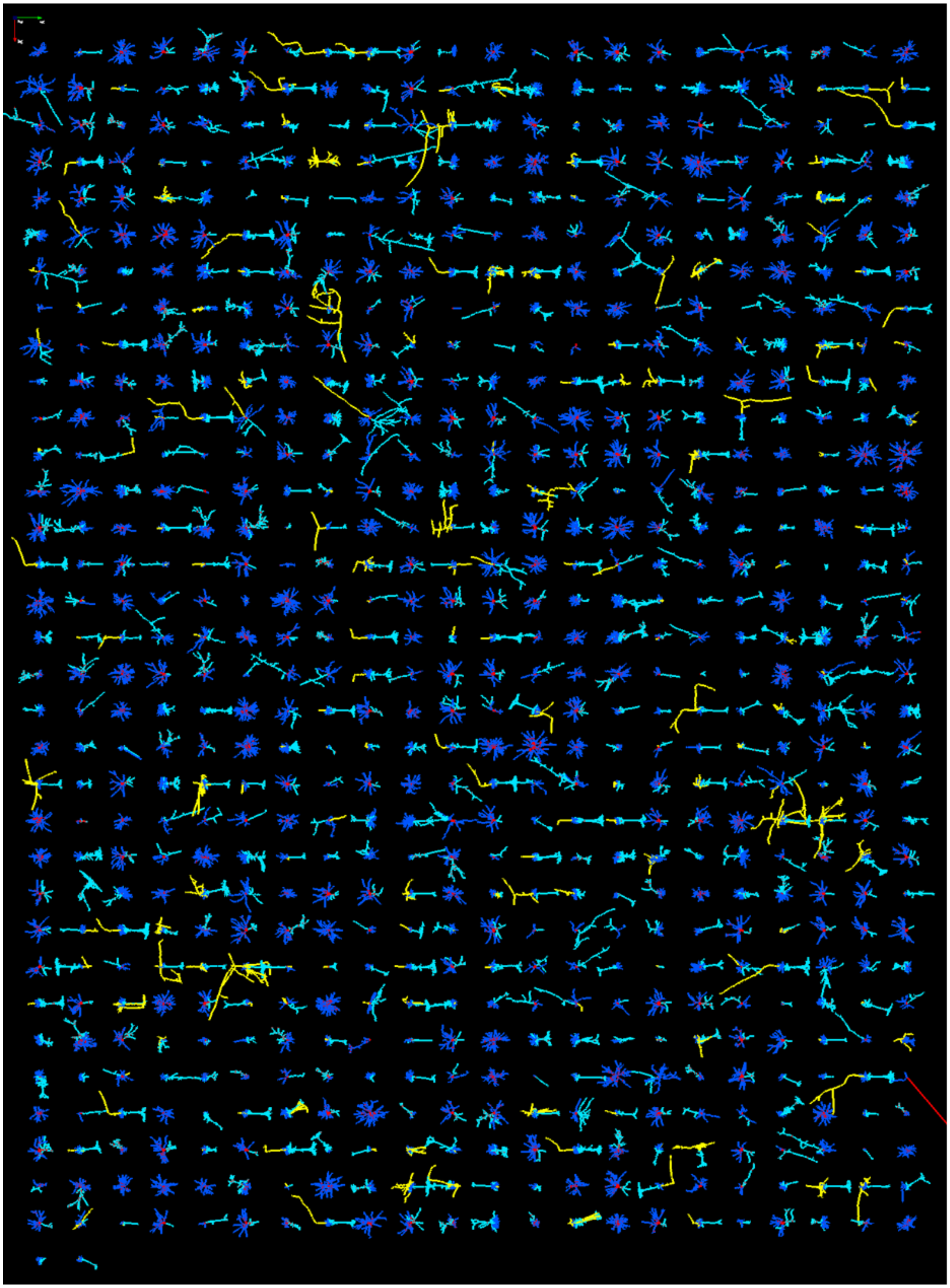
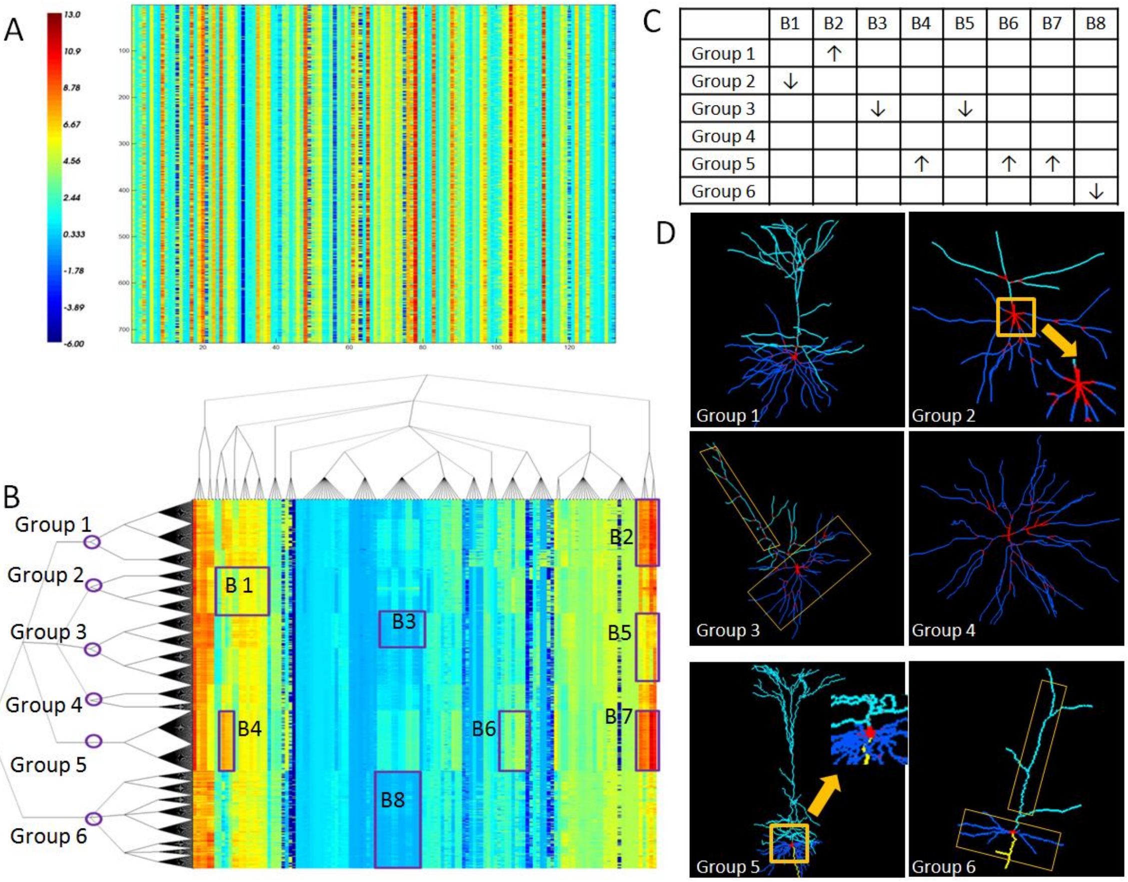


Figure 16. Thumbnail panel of 728 neuron reconstructions in the NeuroMorpho database representing all reconstructed pyramidal cells in the mouse neocortex, containing 6 subclasses of pyramidal cells with varying morphological characteristics.

Since the dataset we used is manually labeled, a confusion matrix was generated as in Figure 22(B). Precision and Recall statistics were computed for each cluster of cells for evaluation. The overall accuracy is also listed in the table.

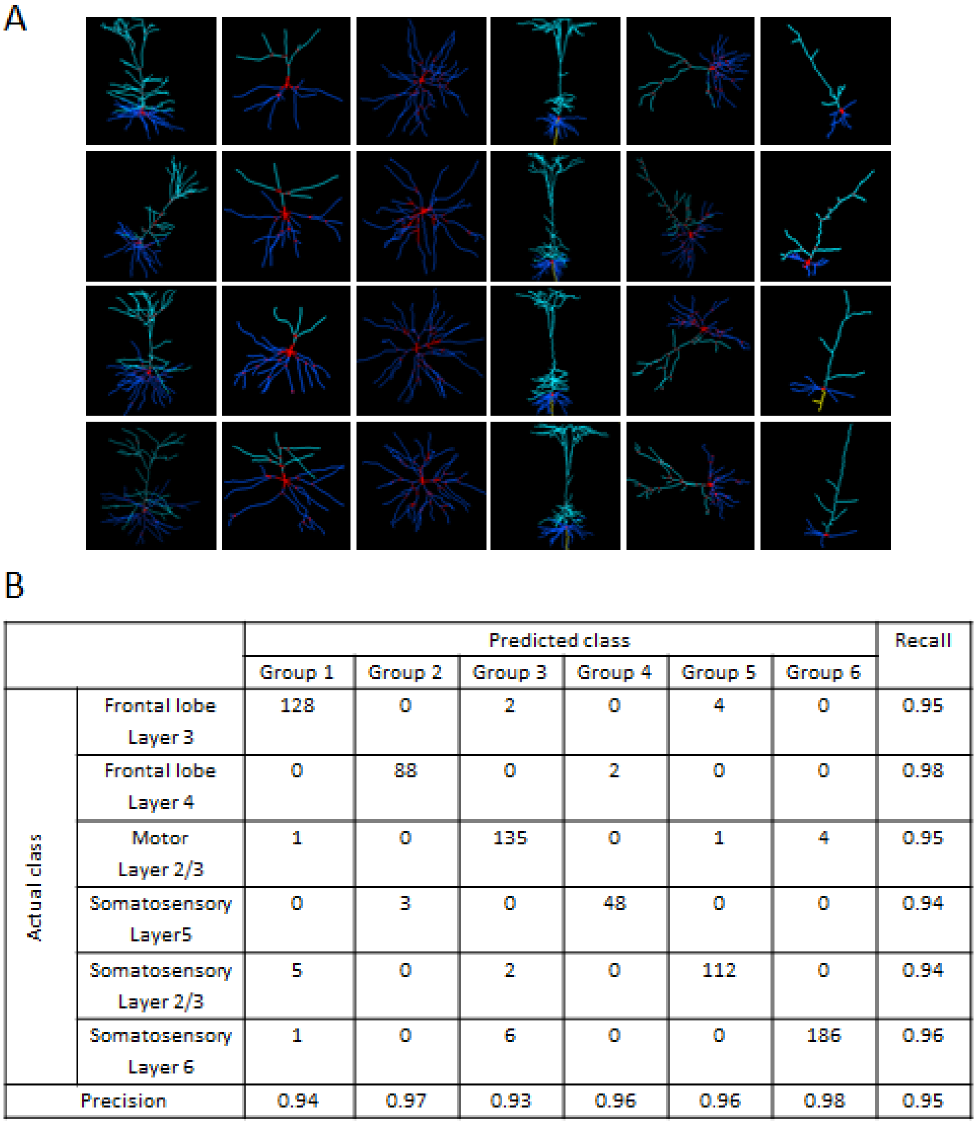
Feature groups with high intra-cluster correlation are of obvious interest for data interpretation. A correlated group of features provide guidance for feature selection and further analysis, and help identify the major features that distinguish the cell clusters. Examples of feature groups that differentiate cell clusters are shown in Figure 21(B, C&D). In Figure 21 (B), blocks that highlighted by purple boxes are feature groups that differentiate a specific cell type from others. In Figure 21(C), the distinguishing features of the groups are highlighted for each block, and the corresponding reconstructions are shown in Figure 21(D). In Group 1 and Group 5, cells are morphologically more complex than other cell groups, correspondingly, in block B2 and B7, the overall complexity features exhibit higher than average values. Cell somas in Group 2 exhibit simpler morphological characteristics compared to other cell groups, and correspondingly, block B1 has lower than average values. Block B1 includes soma size and complexity features such as soma height, soma width, soma depth, soma surface area, soma radii, root branch point. On the other hand, Group 5 has more complex somas, especially root branch point (as B4). In Group 5, the cells have more stems, segments, and branches, as indicated in block B6. Cells in Group 3 and Group 6 are not morphologically symmetric, thus the



	B1	B2	B3	B4	B5	B6	B7	B8
Group 1		↑						
Group 2	↓							
Group 3			↓		↓			
Group 4								
Group 5				↑		↑	↑	
Group 6								↓

Figure 21. Analysis of all 728 pyramidal cell. (A) Data matrix displayed as a heat map. (B) Harmonic co-clustering results. (C, D) Visual confirmation of sample cells from the boxed regions.

feature blocks B3 and B8 have lower than average values. Symmetric features include skewness along the x, y, & z axes, and the skewness magnitude. Additional details about the arbor features are available at the FARSIGHT websiten http://farsight-toolkit.org/wiki/L_Measure_functions. To be specific, in Figure 17 Analysis of all 728 pyramidal cell reconstructions from the mouse neocortex in the NeuroMorpho database. (A) Data matrix of 728 cells and 130 features/cell displayed as a heat map. (B) Harmonic co-clustering results with overlaid circles indicating the six identified subclasses of pyramidal cells. Overlaid boxes B1 - B8 highlight distinctive feature groups of automatically identified cells. (C, D) Visual confirmation of sample cells from the boxed regions. ↑indicates feature values above the average and ↓ indicates feature values below average. (B1: Soma size and complexity features; B2, B5, B7: Overall size and complexity features; B3, B8: Overall symmetric features; B4: Soma size features B6: Segments and branches).



		Predicted class						Recall
		Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	
Actual class	Frontal lobe Layer 3	128	0	2	0	4	0	0.95
	Frontal lobe Layer 4	0	88	0	2	0	0	0.98
	Motor Layer 2/3	1	0	135	0	1	4	0.95
	Somatosensory Layer5	0	3	0	48	0	0	0.94
	Somatosensory Layer 2/3	5	0	2	0	112	0	0.94
	Somatosensory Layer 6	1	0	6	0	0	186	0.96
Precision		0.94	0.97	0.93	0.96	0.96	0.98	0.95

Figure 22. Automatically identified pyramidal sub-types (one subtype in each column in A), and confusion matrix (manually validated) in B for the dataset in Figure 6. The high recall and precision values demonstrate the effectiveness of the proposed method.

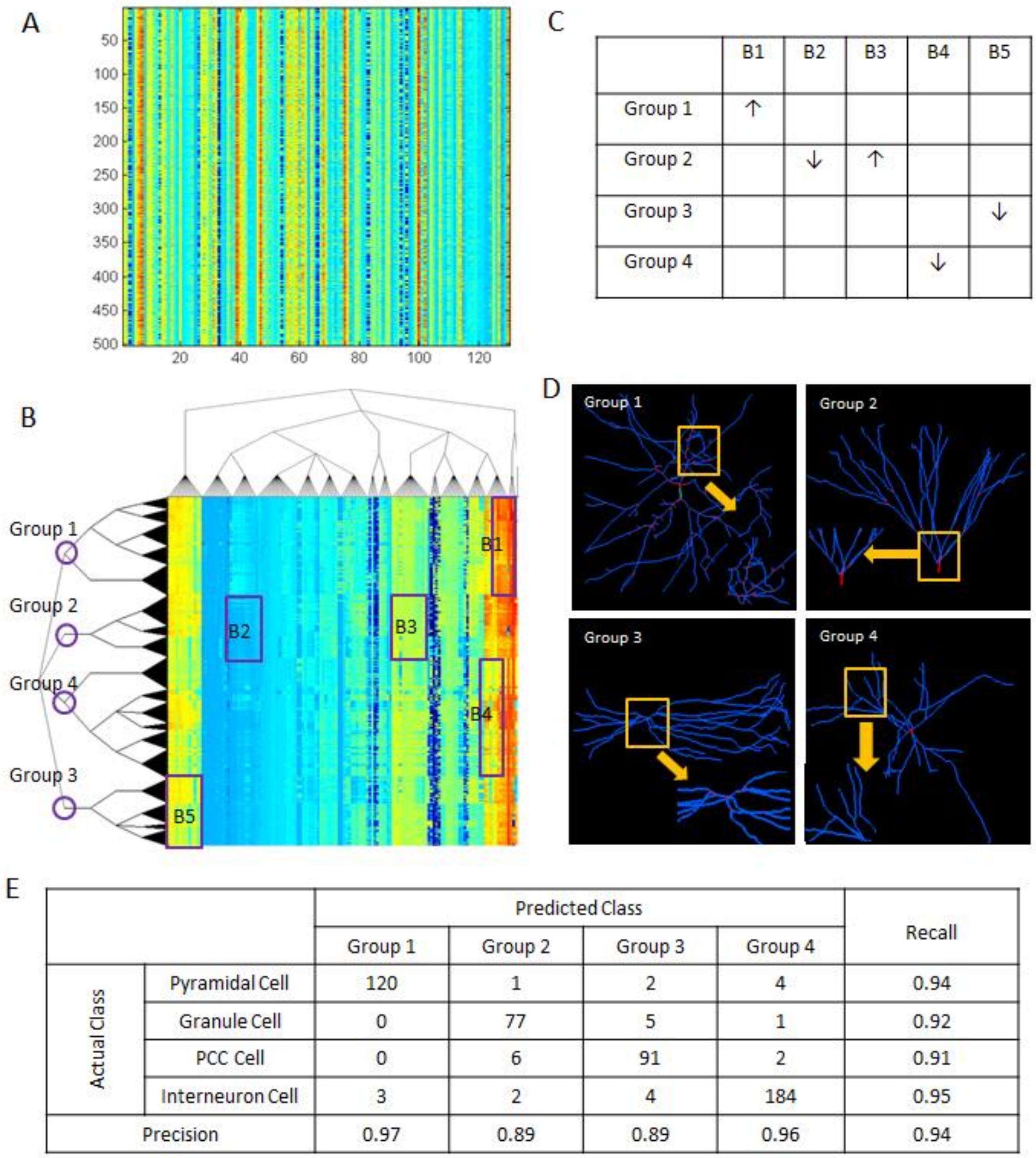
7.4.2 Rat Hippocampal Neurons

This dataset consists of 502 complete neuronal reconstructions of four types of neurons from the rat hippocampus in the NeuroMorpho.Org (Ascoli et al., 2007) database (www.neuromorpho.org). This is a very heterogeneous collection of cells that are imaged under a variety of conditions (e.g., cultured, obtained from neonates, or drug treated), as part of diverse studies including: pyramidal cells from (Ishizuka et al., 1995), pyramidal cells from (Scorza et al., 2011), pyramidal cells from (Tamamaki et al., 1991), granule cells from (Bausch et al., 2006), granule cells from (Carnevale et al., 1997), granule cells from (Arisi et al., 2007), PCC cells and pyramidal cells from (Gulyás et al., 1999), interneuron cells from (Chitwood et al., 1999), and interneuron cells from (Golding et al., 2005).

Despite this heterogeneity, four groups of neurons were correctly identified by the co-clustering algorithm based only on the arbor morphologies, as shown in Figure 23. The four cell types are pyramidal cells, granule cells, PCC cells, and interneurons respectively. A confusion matrix is also given based on the manual labeling in Figure 23 (E). The co-clustering heat map and tree structures for this dataset are shown in Figure 23 (B).

The pyramidal cells, identified as Group 1 in the co-clustering heatmap, have high overall complexity features. For example, the number of segments of each cell is higher than that in other groups, number of branches spread out from each stem is also more than that in other groups. Cells in this group also have higher values for bifurcation values (bifurcation angles, bifurcation diameter ratios, etc). The second group identified by the co-clustering algorithm is the granule group, in which cells have simple morphological structures. Number of branches spread out from each stem is quite low, approximately zero as there is rarely any bifurcations. However, the skewness features of this group are high since all branches are located on one side of the soma. The PCC cells, identified in the third group, also have simple morphological structures. Therefore, the overall complexity features are of low values. On the other hand, it's skewness features are lower than that in the second group though their stems and braches are not symmetric, but distributed uniformly on the surface of the soma. Specifically, in Figure 23, we analyzed of all 502 neuronal cell reconstructions from the rat hippocampus region. (A) Data matrix of 502 cells and 130 features/cell displayed as a heat map. (B) Harmonic co-clustering results with overlaid circles indicating the four identified groups of cells. Overlaid boxes B1 – B5 highlight distinctive feature groups of automatically identified cells. (C, D) Visual confirmation of sample cells from the boxed regions. ↑indicates feature values above the average and ↓ indicates feature values below average. (B1: Overall size and complexity features. B2: Overall symmetric features. B3: Segments and branches. B4: Tips amplitudes. B5: Soma complexity features). (E) shows the confusion matrix using labels specified in the database.

To test the sensitivity of the proposed algorithm to subtle arbor differences, we selected four subtypes of interneurons including bitufted cells, Calbindin (CB) containing cell, Calretinin (CR) containing cells, and Cholecystokinin (CCK) containing cells. The morphologies of these four subgroups are quite similar to each other (especially among the last three subtypes). The algorithm gives an overall accuracy of 79%. Notice that the accuracy is not as high as the other datasets and reflects the difficulty of the task – these four cell subtypes are very difficult to discern visually.



	B1	B2	B3	B4	B5
Group 1	↑				
Group 2		↓	↑		
Group 3					↓
Group 4				↓	

		Predicted Class				Recall
		Group 1	Group 2	Group 3	Group 4	
Actual Class	Pyramidal Cell	120	1	2	4	0.94
	Granule Cell	0	77	5	1	0.92
	PCC Cell	0	6	91	2	0.91
	Interneuron Cell	3	2	4	184	0.95
Precision		0.97	0.89	0.89	0.96	0.94

Figure 23. Analysis of all 502 neuronal cell reconstruction. (A) Data matrix of 502 cells and 130 features, (B) Harmonic co-clustering results, (C, D) Visual confirmation of sample cells from the boxed regions, (E) Confusion matrix using labels specified in the database.

7.5 Binge Alcohol Study

In this section, we present an application of our proposed framework in a binge alcohol study. A brief introduction of the experiment and data acquisition will be given in Section

7.5.1. In Section 7.5.2, we will describe the details of applying the framework, including data preparation, parameter setting up, and analysis results interpretation.

7.5.1 Study & Data Description

It is known that exercise benefits the brain, in part because it powerfully promotes glial health and plasticity. However, exposure to drugs of abuse has been shown to limit future experience-induced brain change. In an experiment conducted by Emily et. al (2014), they investigated whether binge alcohol exposure would reduce the beneficial effects of exercise on glia. They focused on the medial prefrontal cortex (mPFC), an alcohol-vulnerable region, dysfunctions in which contribute to the perpetuation of addiction. Rats underwent a four-day binge exposure, followed by a week of rest and then 4 weeks of exercise. Immunofluorescence was used to label microglia, astrocytes, and neurons in serial tissue sections through the mPFC. The Confocal microscope images were processed using FARSIGHT, a computational image analysis toolkit.

Tissues were collected from a previous study of binge damage and exercise-driven repair in female rats (Maynard and Leasure, 2013b). Experimental rats were given an ethanol diet or an isocaloric control diet every eight hours for four days by intragastric gavage. Seven days after the last dose of alcohol or isocaloric diet, rats in the exercise groups were given access to exercise wheels for a total of five and a half hours each day for four weeks. After 28 days of exercise, rats were killed and their brains were removed for imaging. Three 50 µm serial sections from the mPFC of each animal were processed using multi-channel immunofluorescence to label cell nuclei (4',6-diamidino-2-phenylindole, DAPI), microglia (Iba1), neurons (NeuN), and astrocyte fibrils (GFAP). A Leica SP8 confocal microscope was used to image eight fields of view (hereafter referred to as “tiles”) in a 2x4 rectangle covering a 775 μm x 1,550 μm wide region for each tissue section, resulting in 24 fields of view for each animal. After image acquisition, each Z-stack was separated by channel and saved in TIFF format for processing in FARSIGHT. In the Farsight Pipeline, an initial segmentation step segmented all nucleus in the DAPI channel. A following active learning classification algorithm allows a user interactive classifying of nucleus into different cell types. Furthermore, the astrocytes and microglia arbor reconstructions are derived by using a tracing and reconstruction algorithm.

7.5.2 Analysis on Microglial and Astrocytes

From the above procedures, the total number of microglial cell construction collected in this experiment is 11,822, each with 132 arbor measurements. A table of 11,822 ×132 is set as input to the proposed frame work. We use a default setting of 50 cell clusters and 32 feature groups for the co-clustering algorithm. In Figure 24, we show a heatmap of the reorganized data using the co-clustering algorithm. Population distribution of the four clusters is also given for each experimental condition in Figure 25. There are four cell clusters (ramified, elongated, activated and amoeboid) identified. We present them here in a decreasing order of the cell morphology complexity. A zoomed-in view of the four clusters are shown in Figure 26 (e, f, g, h), selected from a collection of the microglial channel of the four experimental groups in (a, b, c, d). It is clear that the population of amoeboid microglia in the Exercise Binge is significantly higher than in the rest of the experimental conditions.

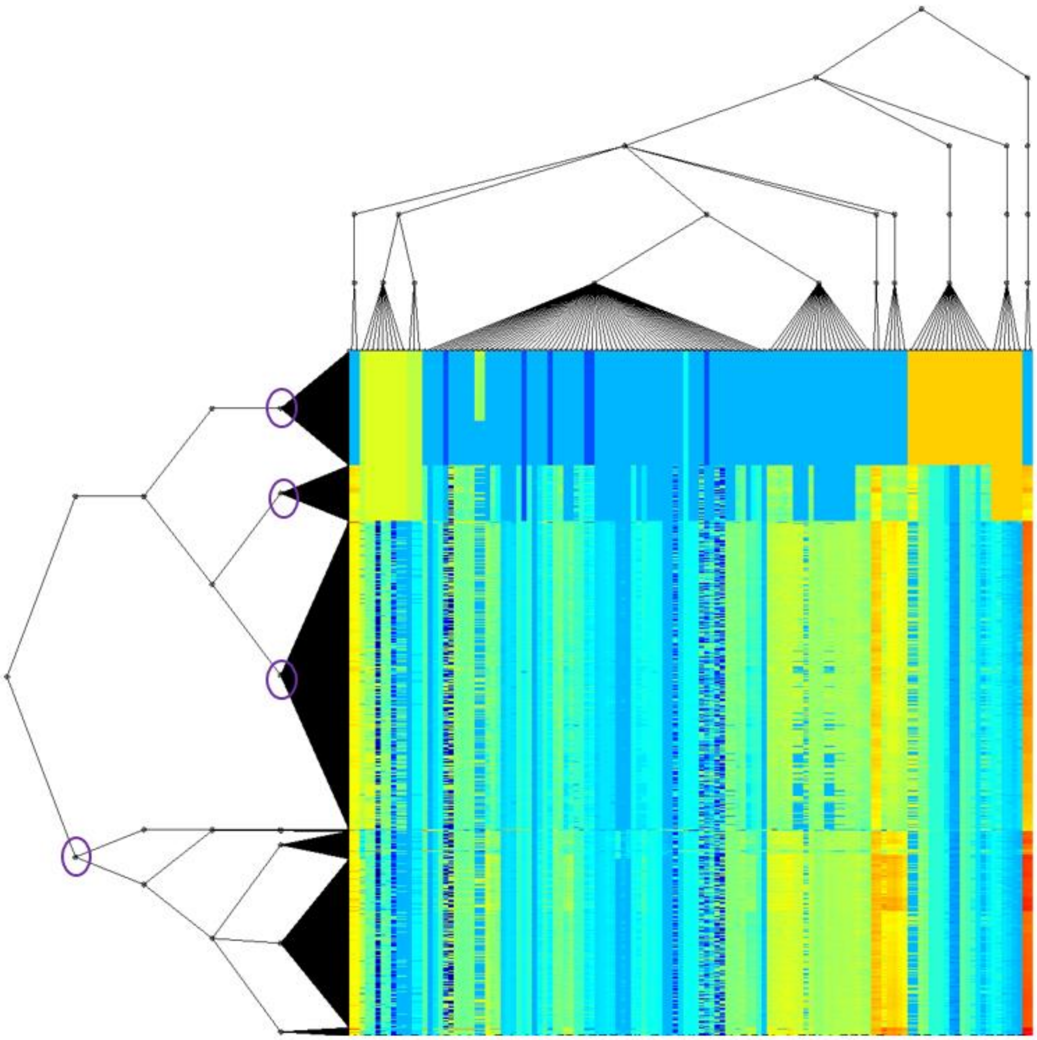


Figure 24. Microglia co-clustering heatmap, rows correspond to cells, columns correspond to measurements. Four clusters are identified from the heatmap.

In the next step, we performed the sparse group lasso. Label information are derived from the co-clustering result. Specifically, in this application, we labeled the entire dataset as 12 classes. Each class represents a combination of cell cluster (3) and experimental condition (4). The reason that the forth cell cluster (amoeboid) is excluded is because these cells are arbor less. Our feature group information in this application are given according to their functional and morphological properties by biologists. The 132 measurements are divided into 32 groups. The group lasso results are presented in Figure

4. In this experiments, our lambda is chose to be in the range of [0, 2.5]. Figure 27 (A) is a coefficient plots across lambdas, the degree of freedom are given when each feature get selected. The MSE is shown in Figure 27 (B), the estimation accuracy increase along while the sparsity decrease. The acceptable range is given between the two dash lines. Further we selected a coefficient vector with degree of freedom 28, which in this case fits best of the regression. We plotted the distribution of each of the significant features across clusters and experimental conditions. The statistics of the high ranking features are given in Table 10. We choose the same parameter and model for further batch learning.

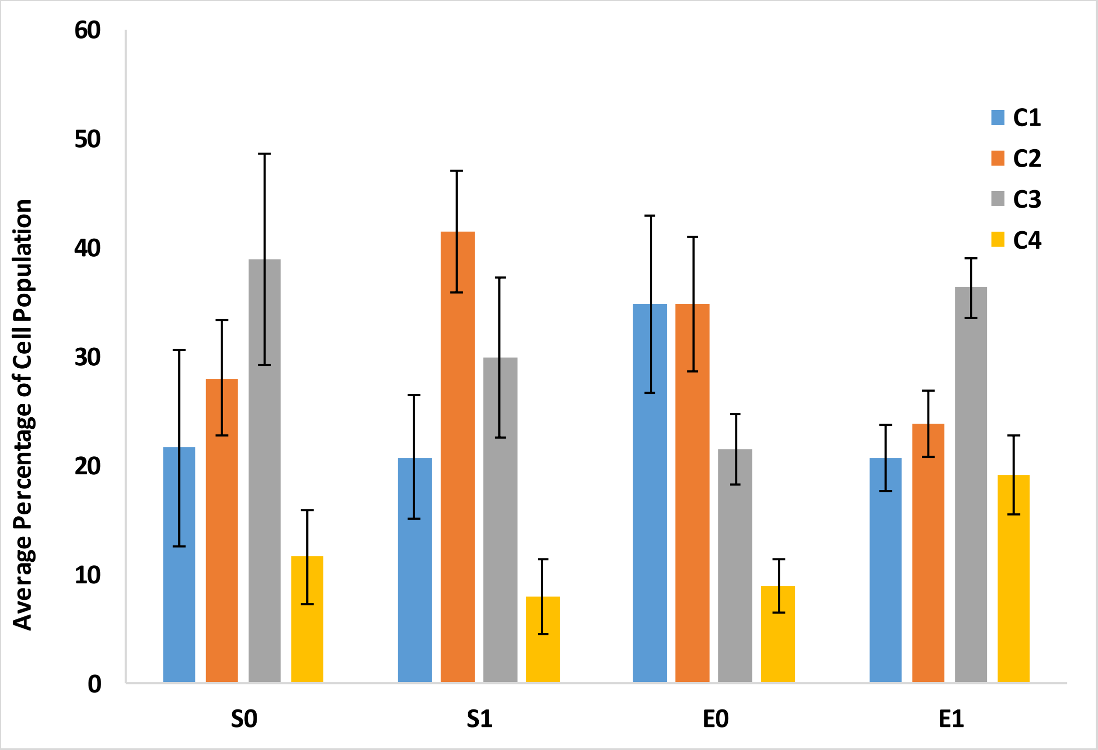


Figure 25. Microglia cell cluster distribution among experiment conditions. S0-sendetary control, S1-sendetary binge,E0- exercise control, and E1-exercise binge. C1, C2, C3 and C4 represent four clusters.

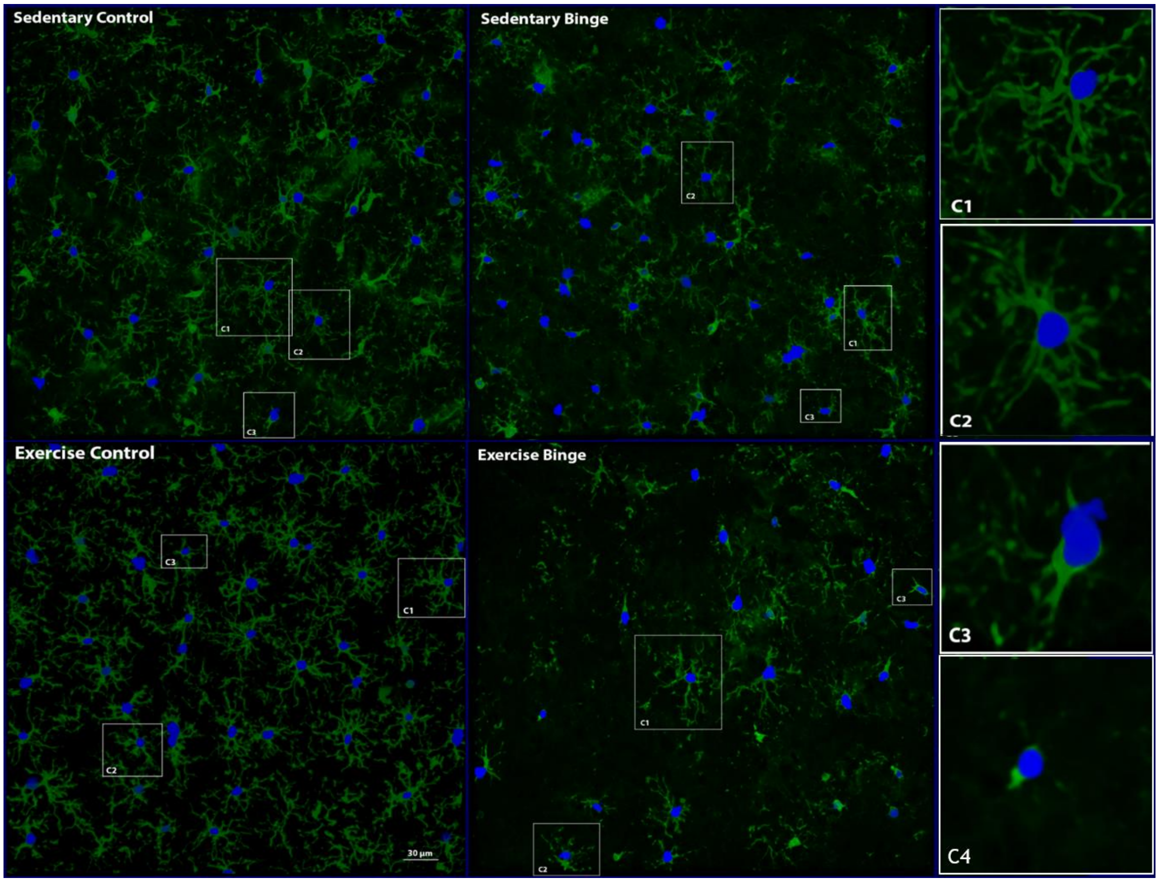


Figure 26. Sample view of microglia sub-types across experimental conditions. C1-ramified, C2-elongated, C3-activated, and C4-amoeboid.

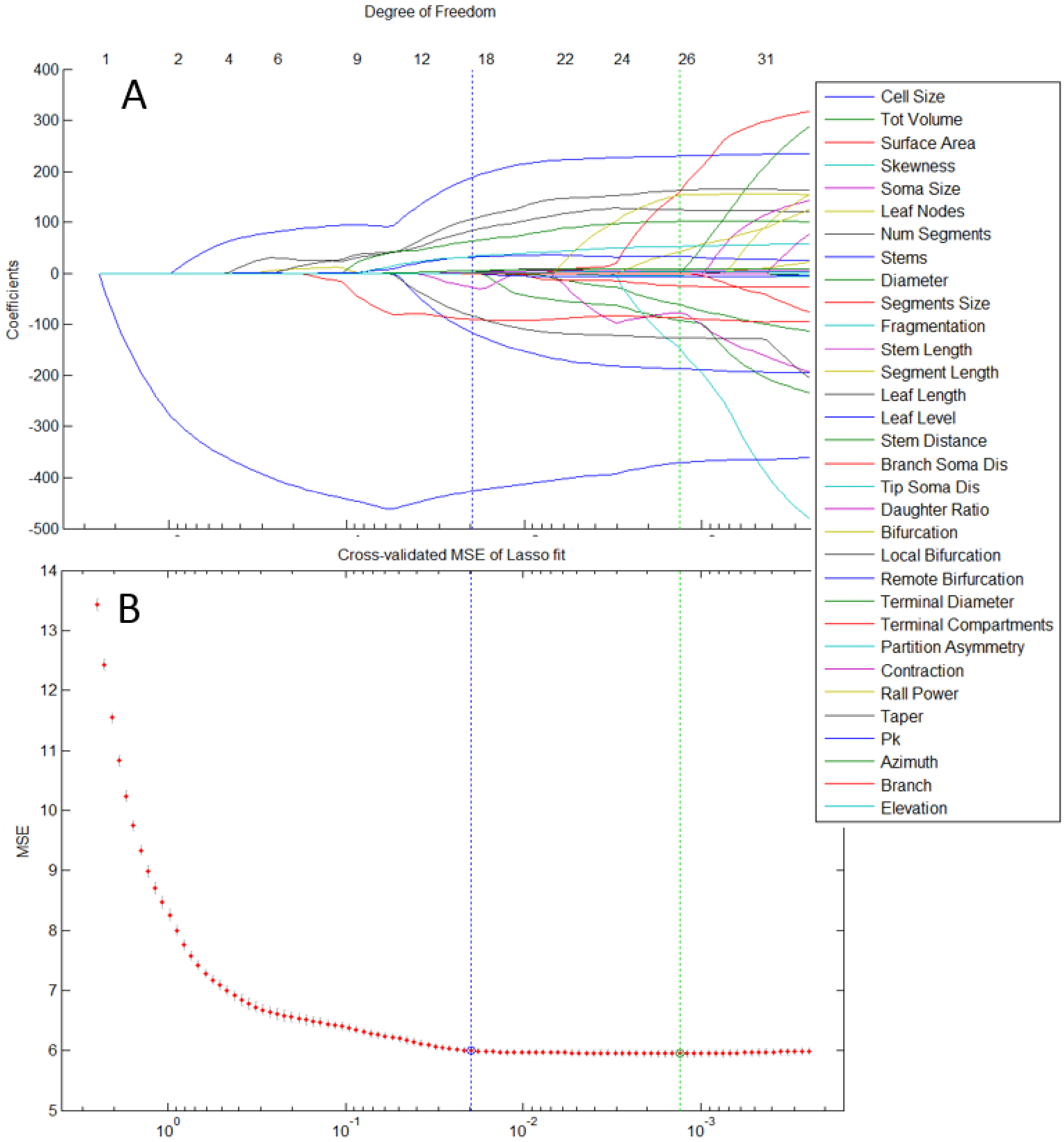


Figure 27. Lasso plot. The x-axis is the logarithm of lamda, the y-axis is the coefficients and mean square error, in (A) and (B), respectively. In (A), each curve represent a feature as listed in the legend. The degree of freedom is given on top when more features been selected. The acceptable range is indicated between the two vertical dash lines.

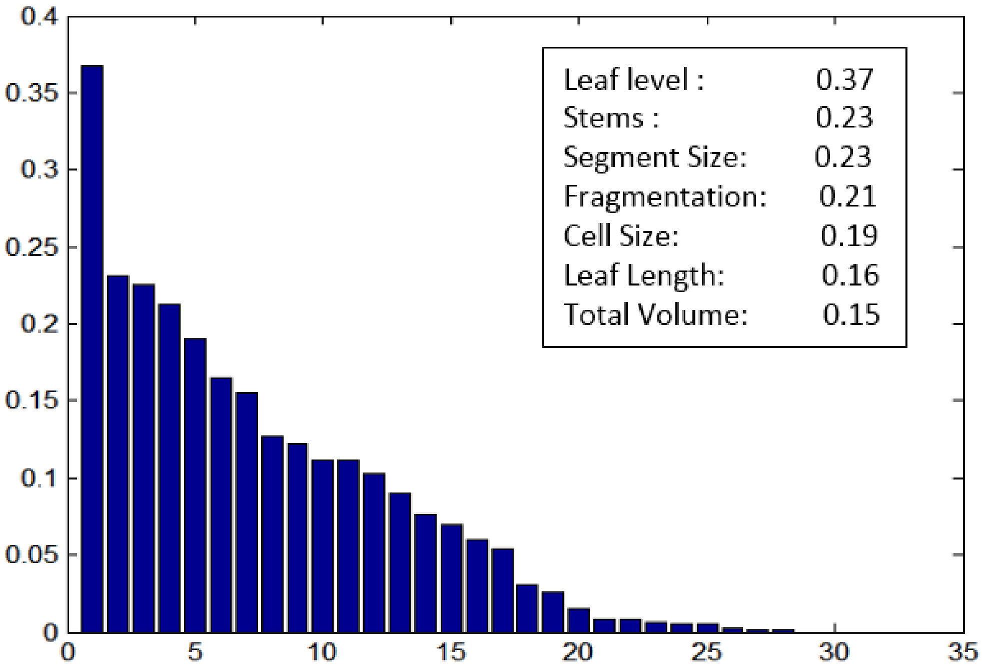


Figure 28. A coefficient vector selected from Figure 27, with degree of freedom 28. The x-axis represent feature index and the y-axis represent coefficient.

Table 10. Examples (cell size & leaf level) of feature distribution across experimental condition and cell cluster.

Cell Size				
	Sedentary Control	Sedentary Binge	Ex Control	Ex Binge
Group 1	328.94 +/- 120.19	281.5 +/- 71.53	252.3 +/- 90.53	157.77 +/- 71.57
Group 2	192.98 +/- 40.23	189.63 +/- 37.91	147.69 +/- 33.02	130.44 +/- 31.19
Group 3	92.69 +/- 36.87	92.75 +/- 39.5	76.59 +/- 28.74	66.77 +/- 26.63
Leaf Level				
Group 1	6.58 +/- 2.57	5.64 +/- 1.28	5.11 +/- 2.09	3.54 +/- 3.03
Group 2	3.98 +/- 0.85	3.84 +/- 0.83	3 +/- 0.78	2.41 +/- 0.85
Group 3	1.92 +/- 0.83	1.87 +/- 0.83	1.5 +/- 0.62	1.39 +/- 0.69

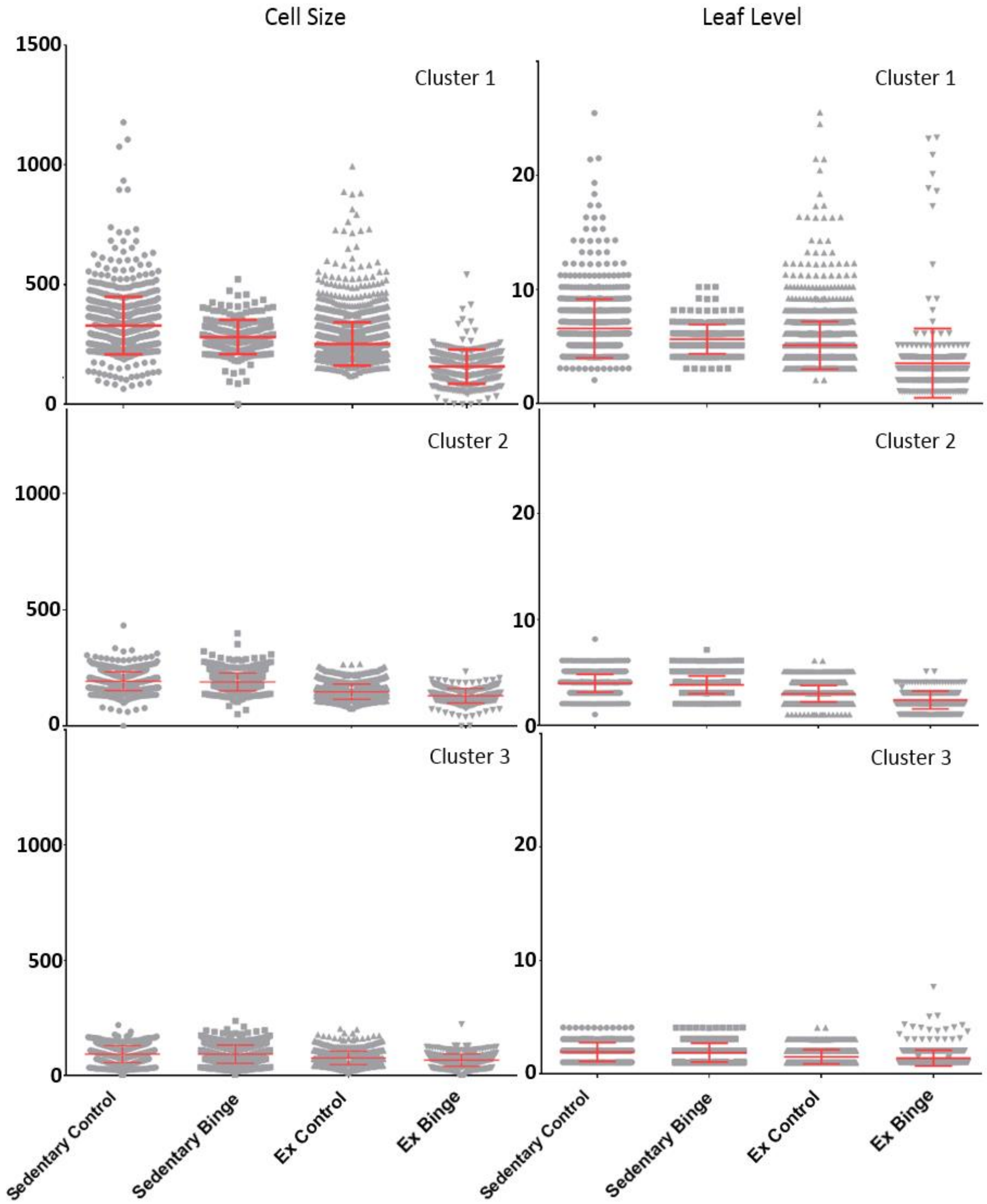


Figure 29. Distribution of Cell Size and Leaf Level across experimental groups and cell clusters.

Chapter 8 Conclusion

In this dissertation, we proposed a comprehensive and usable method for population-scale comparative arbor analytics that includes three major components. First, we use our previously reported unsupervised harmonic co-clustering algorithm for identifying groups of cells with similar morphologies in a hierarchical manner, and simultaneously identifying the hierarchical grouping patterns among the corresponding arbor measurements. This algorithm is applied to the combined ensemble. For example, if we are interested in comparative profiling of two cell populations A and B, harmonic co-clustering is performed on the combination of A and B. This step can be thought of as a grand reorganization of the data matrix that allows us to establish a common baseline of cell grouping patterns, and the underlying feature groups for the combined dataset, enabling a sensitive analysis of the differences between A and B in subsequent steps.

The unsupervised hierarchical harmonic co-clustering algorithm is a practically effective tool for quantitative arbor analytics, and in our experience, requires very little effort in terms of settings and parameter tuning. The static and low-resolution figures in the paper do not fully convey the live interactive power of this method as can be experienced on a large dual screen monitor. For example, the names of the L-measure parameters are visible in the live system but not on our figures since the text become too small. Given a large table of L-measure data, our algorithm identifies groups of similar cells with similar features. The investigator can then study a small number of sample cells from each group knowing that they typify members of the group. Furthermore, our method makes it convenient to compare groups of cells visually and quantitatively. It's linked embedding into the FARSIGHT trace viewing and editing tool enables efficient visual confirmation of the analyses. Our work has also resulted in a modest but useful extension to the L-measure. The co-clustering algorithm is agnostic to the choice of arbor measurements, and other sets of arbor measurements can be used just as effectively.

This method is robust to noise, produces biologically meaningful yet concise results, and is scalable to large datasets. The algorithm outperforms recent methods in dealing with high dimensional datasets due to the adoption of the diffusion distance measure. By coupling the geometry of row space and column spaces, it is not only able to extract the cluster structures in row and column spaces, but also map the sub-block structure of the data matrix.

The Haar wavelet-based smoothing provides a theoretically sound and practically effective way for co-clustering large datasets. For small datasets (<50 data points), the proposed method will produce valid results, but not offer any significant benefits over conventional co-clustering methods. There is no theoretical upper bound to our method, and it should, in principle, work correctly given enough computing resources. With such future applications in mind, our C++ software implementation has been written in a manner that is 64-bit compliant using long word lengths, in order to handle potentially much larger datasets in the future. To date, we have conducted experiments with ~30,000 data points. Beyond this size, the user will likely be limited by the available visualization resources (e.g., screen size).

The strengths and limitations of the proposed method are ultimately tied to the expressiveness of the data matrix fed to the harmonic co-clustering. It is critical to demonstrate the utility and robustness of the harmonic co-clustering analysis with large heterogeneous populations of cells, and the ability to differentiate cell subtypes with subtle morphological differences. For this purpose, we have focused our efforts on two such datasets in the NeuroMorpho database, one of which is to identify different cell types from the same region, while the other is to identify subtypes of pyramidal cells. Similarly, the algorithm should be able to cluster any arbor cells when similar measurements are computed. The performance of the clustering will depend on the quality of the reconstructions, and the sensitivity with which subtle arbor differences between different neuronal subtypes are reflected in the quantitative L-measure data. In other words, the co-clustering algorithm is able to group cell subtypes as long as the morphological differences are captured with sufficient sensitivity by the feature measurements. We expect that libraries like the L-measure will continue to be refined in the future, and the co-clustering algorithm will be able to exploit these refinements. Some of these refinements may well be inspired by quantitative arbor analytics on a population scale.

References

Amershi, S., & Conati, C. (2009). Combining Unsupervised and Supervised Classification to Build User Models for Exploratory. *JEDM-Journal of Educational Data Mining*, 1(1), 18-71.

Arisi, G. M., & Garcia-Cairasco, N. (2007). Doublecortin-positive newly born granule cells of hippocampus have abnormal apical dendritic morphology in the pilocarpine model of temporal lobe epilepsy. *Brain research,* 1165, 126-134.

Ascoli, G. A., Donohue, D. E., & Halavi, M. (2007). NeuroMorpho. Org: a central resource for neuronal morphologies. *The Journal of Neuroscience*, 27(35), 9247-9251.

Banerjee, A., Dhillon, I., Ghosh, J., Merugu, S., & Modha, D. S. (2004, August). A generalized maximum entropy approach to bregman co-clustering and matrix approximation. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 509-514). ACM.

Bausch, S. B., He, S., Petrova, Y., Wang, X. M., & McNamara, J. O. (2006). Plasticity of both excitatory and inhibitory synapses is associated with seizures induced by removal of chronic blockade of activity in cultured hippocampus. *Journal of Neurophysiology,* 96(4), 2151.

Balasubramanian, M., & Schwartz, E. L. (2002). The isomap algorithm and topological stability. *Science*, 295(5552), 7-7.

Carnevale, N. T., Tsai, K. Y., Claiborne, B. J., & Brown, T. H. (1997). Comparative electrotonic analysis of three classes of rat hippocampal neurons. *Journal of Neurophysiology,* 78(2), 703-720.

Chitwood, R. A., Hubbard, A., & Jaffe, D. B. (1999). Passive electrotonic properties of rat hippocampal CA3 interneurones. *The Journal of Physiology,* 515(3), 743-756.

Chen, G., Sullivan, P. F., & Kosorok, M. R. (2013). Biclustering with heterogeneous variance. *Proceedings of the National Academy of Sciences,* 110(30), 12253-12258.

Chen, C. C., Abrams, S., Pinhas, A., & Brumberg, J. C. (2009). Morphological heterogeneity of layer VI neurons in mouse barrel cortex. *Journal of Comparative Neurology,* 512(6), 726-746.

Cheng, Y., & Church, G. M. (2000, August). Biclustering of expression data. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology* (Vol. 8, pp. 93-103).

Coifman, R. R., & Gavish, M. (2011). Harmonic analysis of digital data bases. In *Wavelets and Multiscale Analysis* (pp. 161-197).

Coifman, R. R., & Lafon, S. (2006). Diffusion maps. *Applied and Computational Harmonic Analysis,* 21(1), 5-30.

Coifman, R. R., & Maggioni, M. (2006). Diffusion wavelets. *Applied and Computational Harmonic Analysis,* 21(1), 53-94.

Coombs, J., Van Der List, D., Wang, G. Y., & Chalupa, L. M. (2006). Morphological properties of mouse retinal ganglion cells. *Neuroscience,* 140(1), 123-136.

Cuntz, H., Forstner, F., Haag, J., & Borst, A. (2008). The morphological identity of insect dendrites. *PLoS Computational Biology*, 4(12), e1000251.

Dhillon, I. S., Mallela, S., & Modha, D. S. (2003, August). Information-theoretic co-clustering. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 89-98). ACM.

Ding, C., He, X., & Simon, H. D. (2005, April). On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proc. SIAM Data Mining Conf* (No. 4, pp. 606-610).

Eren, K., Deveci, M., Küçüktunç, O., & Çatalyürek, Ü. V. (2013). A comparative analysis of biclustering algorithms for gene expression data. *Briefings in Bioinformatics*, 14(3), 279-292.

Gao, B. J., Griffith, O. L., Ester, M., & Jones, S. J. (2006, August). Discovering significant opsm subspace clusters in massive gene expression data. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 922-928). ACM.

Gavish, M., Nadler, B., & Coifman, R. R. (2010). Multiscale wavelets on trees, graphs and high dimensional data: Theory and applications to semi supervised learning. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)* (pp. 367-374).

George, T., & Merugu, S. (2005, November). A scalable collaborative filtering framework based on co-clustering. In *Data Mining, Fifth IEEE International Conference* on (pp. 4-pp). IEEE.

Getz, G., Gal, H., Kela, I., Notterman, D. A., & Domany, E. (2003). Coupled two-way clustering analysis of breast cancer and colon cancer gene expression data. *Bioinformatics,* 19(9), 1079-1089.

Golding, N. L., Mickus, T. J., Katz, Y., Kath, W. L., & Spruston, N. (2005). Factors mediating powerful voltage attenuation along CA1 pyramidal neuron dendrites. The *Journal of Physiology,* 568(1), 69-82.

Gulyás, A. I., Megías, M., Emri, Z., & Freund, T. F. (1999). Total number and ratio of excitatory and inhibitory synapses converging onto single interneurons of different types in the CA1 area of the rat hippocampus. *The Journal of Neuroscience,* 19(22), 10082-10097.

Halavi, M., Hamilton, K. A., Parekh, R., & Ascoli, G. A. (2012). Digital reconstructions of neuronal morphology: three decades of research trends. *Frontiers in Neuroscience, 6*.

Hartigan, J. A. (1972). Direct clustering of a data matrix. *Journal of the American Statistical Association,* 67(337), 123-129.

Ho, S. Y., Chao, C. Y., Huang, H. L., Chiu, T. W., Charoenkwan, P., & Hwang, E. (2011). NeurphologyJ: An automatic neuronal morphology quantification method and its application in pharmacological discovery. *BMC Bioinformatics,* 12(1), 230.

Hochreiter, S., Bodenhofer, U., Heusel, M., Mayr, A., Mitterecker, A., Kasim, A., & Clevert, D. A. (2010). FABIA: factor analysis for bicluster acquisition. *Bioinformatics, 26*(12), 1520-1527.

Ishizuka, N., Cowan, W. M., & Amaral, D. G. (1995). A quantitative analysis of the dendritic organization of pyramidal cells in the rat hippocampus. *Journal of Comparative Neurology,* 362(1), 17-45.

Jinushi-Nakao, S., Arvind, R., Amikura, R., Kinameri, E.,Liu, A. W., & Moore, A. W. (2007). Knot/Collier and cut control different aspects of dendrite cytoskeleton and synergize to define final arbor shape. *Neuron,* 56(6), 963-978.

Kulkarni, P. M., Barton, E., Savelonas, M., Padmanabhan, R., Lu, Y., Trett, K., & Roysam, B. (2015). Quantitative 3-D Analysis of GFAP Labeled Astrocytes from Fluorescence Confocal Images. *Journal of Neuroscience Methods*, 246, 38-51.

Kullback, S., Leibler, R.A. (1951). On Information and Sufficiency. *Annals of Mathematical Statistics* 22 (1): 79–86.

Kozlowski, C., & Weimer, R. M. (2012). An automated method to quantify microglia morphology and application to monitor activation state longitudinally in vivo. *PLoS One,* 7(2), e31814.

Krieger, P., Kuner, T., & Sakmann, B. (2007). Synaptic connections between layer 5B pyramidal neurons in mouse somatosensory cortex are independent of apical dendrite bundling. *The Journal of Neuroscience,* 27(43), 11473-11482.

Lafon, S. S. (2004). Diffusion maps and geometric harmonics (Doctoral dissertation, Yale University).

Lu, Y., Trett, K., Shain, W., Carin, L., Coifman, R., & Roysam, B. (2013, April). Quantitative profiling of microglia populations using harmonic co-clustering of arbor morphology measurements. *In Biomedical Imaging (ISBI), 2013 IEEE 10th International Symposium on* (pp. 1360-1363). IEEE.

Lu, Y., Carin, L., Coifman, R., Shain, W., & Roysam, B. (2015). Quantitative arbor analytics: unsupervised harmonic co-clustering of populations of brain cell arbors based on L-measure. *Neuroinformatics*, 13(1), 47-63.

Luisi, J., Narayanaswamy, A., Galbreath, Z., & Roysam, B. (2011). The FARSIGHT trace editor: an open source tool for 3-D inspection and efficient pattern analysis aided editing of automated neuronal reconstructions. *Neuroinformatics*, 9(2), 305-315.

Maynard, M. E., & Leasure, J. L. (2013). Exercise enhances hippocampal recovery following Binge Ethanol exposure. *PloS One*, 8(9), e76644.

Megjhani, M., Rey-Villamizar, N., Merouane, A., Lu, Y., Mukherjee, A., Trett, K., & Roysam, B. (2015). Population-scale three-dimensional reconstruction and quantitative profiling of microglia arbors. *Bioinformatics*, btv109.

Meijering, E. (2010). Neuron tracing in perspective. *Cytometry Part A, 77(7)*, 693-704.

Morrison, H. W., & Filosa, J. A. (2013). Abstract WP331: Diverse Microglia Morphologies Induced By Ischemic Stroke And Reperfusion Are Not Accompanied

By Altered Brain Inducible Nitric Oxide Synthase *Expression. Stroke, 44(Suppl 1)*, AWP331-AWP331.

Padmanabhan, R. K., Somasundar, V. H., Griffith, S. D., Zhu, J., Samoyedny, D., Tan, K. S., & Lee, W. M. (2014). An Active Learning Approach for Rapid Characterization of Endothelial Cells in Human Tumors. *PloS One*, 9(3), e90495.

Parekh, R., & Ascoli, G. A. (2013). Neuronal morphology goes digital: a research hub for cellular and system neuroscience. *Neuron,* 77(6), 1017-1038.

Peng, H., Hawrylycz, M., Roskams, J., Hill, S., Spruston, N., Meijering, E., & Ascoli, G. A. (2015). BigNeuron: large-scale 3D neuron reconstruction from optical microscopy images. *Neuron,* 87(2), 252-256

Peng, H., Long, F., & Myers, G. (2011). Automatic 3D neuron tracing using all-path pruning. *Bioinformatics*, 27(13), i239-i247.

Polavaram, S., Gillette, T. A., Parekh, R., & Ascoli, G. A. (2014). Statistical analysis and data mining of digital reconstructions of dendritic morphologies. *Frontiers in Neuroanatomy,* 8.

Powers, D. M. W. (2011). Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, 2(1), 37-63.

Rey-Villamizar, N. (2014). Large-scale automated image analysis for computational profiling of brain tissue surrounding implanted neuroprosthetic devices using Python. *Frontiers in Neuroinformatics*, 8.

Rocher, A. B., Crimins, J. L., Amatrudo, J. M., Kinson, M. S., Todd-Brown, M. A., Lewis, J., & Luebke, J. I. (2010). Structural and functional changes in tau mutant mice neurons are not linked to the presence of NFTs. *Experimental Neurology,* 223(2), 385-393.

Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science,* 290(5500), 2323-2326.

Santamaría, R., Therón, R., & Quintales, L. (2008). BicOverlapper: a tool for bicluster visualization. *Bioinformatics,* 24(9), 1212-1213.

Scorcioni, R., Polavaram, S., & Ascoli, G. A. (2008). L-measure: a web-accessible tool for the analysis, comparison and search of digital reconstructions of neuronal morphologies. *Nature Protocols,* 3(5), 866-876.

Shang, F., Jiao, L. C., & Wang, F. (2012). Graph dual regularization non-negative matrix factorization for co-clustering. *Pattern Recognition,* 45(6), 2237-2250.

Sibson, R. (1973). SLINK: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal,* 16(1), 30-34.

Simon, N., Friedman, J., Hastie, T., & Tibshirani, R. (2013). A sparse-group lasso. *Journal of Computational and Graphical Statistics,* 22(2), 231-245.

Strömberg, J. O. (1998). Computation with wavelets in higher dimensions. *In Proceedings of the International Congress of Mathematicians* (Vol. 3, pp. 523-532).

Song H, Stevens CF, Gage FH. Astroglia induce neurogenesis from adult neural stem cells. *Nature*. 2002 May 2, 417(6884):39–44.

Tamamaki, N., & Nojyo, Y. (1991). Crossing fiber arrays in the rat hippocampus as demonstrated by three dimensional reconstruction. *Journal of Comparative Neurology,* 303(3), 435-442.

Tan, Y. H., Terrill, S. E., Paranjape, G. S., Stine, K. J., & Nichols, M. R. (2014). The influence of gold surface texture on microglia morphology and activation. *Biomaterials Science,* 2(1), 110-120.

Tang, C., & Zhang, A. (2005). Interrelated two-way clustering and its application on gene expression data. *International Journal on Artificial Intelligence Tools,* 14(04), 577-597.

Trevelyan, A. J., Sussillo, D., Watson, B. O., & Yuste, R. (2006). Modular propagation of epileptiform activity: evidence for an inhibitory veto in neocortex.*The Journal of Neuroscience,* 26(48), 12447-12455.

Van der Maaten, L. J. P., Postma, E. O., & van den Herik, H. J. (2007). Matlab toolbox for dimensionality reduction. *MICC, Maastricht University*.

Verkhratsky A, Butt A. Glial neurobiology. 1st ed. West Sussex, England: John Wiley & Sons, 2007.

Wang, Y., Narayanaswamy, A., Tsai, C. L., & Roysam, B. (2011). A broadly applicable 3-D neuron tracing method based on open-curve snake. *Neuroinformatics,* 9(2-3), 193-217.

Xie, X. L., & Beni, G. (1991). A validity measure for fuzzy clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* 13(8), 841-847.

Xu, X., Lu, Y., Tung, A. K., & Wang, W. (2006, April). Mining shifting-and-scaling co-regulation patterns on gene expression profiles. In *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on* (pp. 89-89). IEEE.

Xu, Y., Savelonas, M., Qiu, P., Trett, K., Shain, W., & Roysam, B. (2013, April). Unsupervised inference of arbor morphology progression for microglia from confocal microscope images. In *Biomedical Imaging (ISBI), 2013 IEEE 10th International Symposium on* (pp. 1356-1359). IEEE.

Yu, K., Zhang, T., & Gong, Y. (2009). Nonlinear learning using local coordinate coding. In *Advances in Neural Information Processing Systems* (pp. 2223-2231).

Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1), 49-67.

Zanier, E. R., Fumagalli, S., Perego, C., Pischiutta, F., & De Simoni, M. G. (2015). Shape descriptors of the “never resting” microglia in three different acute brain injury models in mice. *Intensive Care Medicine Experimental,* 3(1), 1-18.

Zhou, Z., Liu, X., Long, B., & Peng, H. (2016). TReMAP: Automatic 3D neuron reconstruction based on tracing, reverse mapping and assembling of 2D projections. *Neuroinformatics*, 14(1), 41-50.