An approach to clustering biological phenotypes

A Dissertation presented to the Faculty of the Graduate School
University of Missouri-Columbia

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

---

by
Avimanyou Kumar Vatsa

Dr. Toni Kazic, Dissertation Supervisor

July, 2017

The undersigned, appointed by the Dean of the Graduate School, have examined the dissertation entitled

An approach to clustering biological phenotypes

Presented by Avimanyou Kumar Vatsa

A candidate for the degree of Doctor of Philosophy

And hereby certify that in their opinion it is worthy of acceptance.

Dr. Toni Kazic

Dr. Dong Xu

Dr. Jianlin Cheng

Dr. Tony Han

ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisor Dr. Toni Kazic for her motivation, useful comments, remarks and engagement through the learning process of this Ph.D. dissertation. I am very much grateful for her professionalism, valuable guidance and financial support throughout this project and my entire program of study and research. Her guidance helped me throughout the time of research and writing of this dissertation. Furthermore I would like to thank Dr. Ann E. Stapleton for much excellent advice on all sorts of issues. I would also like to thank my other committee members — Dr. Dong Xu, Dr. Jianlin Cheng and Dr. Tony Han — for their guidance towards my Doctor of Philosophy degree. I am greateful to Dr. Shailesh Chandra for providing consistent enthusiasm, encouragement and other supports.

I am very grateful to the National Science Foundation, MCB-1122130, for supporting me through part of this work; and to the USDA NIFA National Research Initiative Competitive Grant, 2009-35100-05028, for supporting Dr. Stapleton's experiments on stress phenotypes. I also express my gratitude to University of Missouri - Columbia for providing graduate research and teaching assistantships.

Last but not the least, I would like to thank my parents, Shobha (wife), Savya (son) and Ishi (daughter) for their time, encouragement, motivation and all the fun we have had during this period.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

abbreviation	*full name*
Δh	Difference in height
Δc	Difference in canopy spread
Δs	Difference in stem diameter
HDD	High-dimensional data
MAD	Median absolute difference
IQR	Inter quantile range
treat1	Growth Condition 1
treat2	Growth Condition 2
treat3	Growth Condition 3
treat4	Growth Condition 4
treat5	Growth Condition 5
treat6	Growth Condition 6
treat7	Growth Condition 7
treat8	Growth Condition 8
treat9	Growth Condition 9

ABSTRACT

Recently emerging approaches to high-throughput phenotyping have become important tools in unraveling the biological basis of agronomically and medically important phenotypes. These experiments produce very large sets of either low or high-dimensional data. Finding clusters in the entire space of high-dimensional data (HDD) is a challenging task, because the relative distances between any two objects converge to zero with increasing dimensionality. Additionally, real data may not be mathematically well behaved. Finally, many clusters are expected on biological grounds to be "natural" — that is, to have irregular, overlapping boundaries in different subsets of the dimensions. More precisely, the natural clusters of the data could differ in shape, size, density, and dimensionality; and they might not be disjoint.

In principle, clustering such data could be done by dimension reduction methods. However, these methods convert many dimensions to a smaller set of dimensions that make the clustering results difficult to interpret and may also lead to a significant loss of information. Another possible approach is to find subspaces (subsets of dimensions) in the entire data space of the HDD. However, the existing subspace methods don't discover natural clusters. Therefore, in this dissertation I propose a novel data preprocessing method, demonstrating that a group of phenotypes are interdependent, and propose a novel density-based subspace clustering algorithm for high-dimensional data, called Dynamic Locally Density Adaptive Scalable Subspace Clustering (DynaDASC). This algorithm is relatively locally density adaptive, scalable, dynamic, and nonmetric in nature, and discovers natural clusters.

1 Introduction

1.1 High-Dimensional Data

In recent years, high-dimensional data are increasing as data collection technologies evolve, and there is a corresponding growth in such datasets. The real-world raw dataset can be gathered from a variety of sources and from different kinds of applications, like research digital photography, surveillance videos, plant phenotyping in the field and laboratory [6,47], and high throughput molecular biology experiments. The key need is to extract knowledge from high-dimensional raw datasets using powerful and effective data analytic methods. One of the best ways to discover knowledge is by grouping variates, since the groups contain many variates that contribute to the outcome variates.

1.1.1 Problems with High-Dimensional Data

Distances Converge to a Uniform Distribution In high-dimensional data space, a full dimensional distance is often no longer meaningful, since the nearest neighbor of a point is expected to be almost as far as its farthest neighbor [9].

As the dimensionality of the data increases, so does the volume of the space around each point in a Voronoi tessellation. As the data become more sparse, the distances among the points converge to a uniform distribution. This problem is often called the "curse of dimensionality" [10, 16, 44, 68, 71, 75, 78, 83]. Similarly, the "density divergence problem" states that different subspace cardinalities have different region densities, as it can be represented by Equation 1 [10].

$$\forall \varepsilon : \lim_{dim \to \infty} P\big(d_{max(o)} < (1+\varepsilon) d_{min(o)}\big) = 1 \tag{1}$$

It states that as the number of dimensions increases, for some object o, the proba-

bility P that the maximum distance between objects, $d_{max(o)}$, is less than that of the minimum distance between objects, $d_{min(o)}$ within some threshold ϵ, converges to 1. It is applicable to different distributions. Thus, increasing data dimensionality results in the loss of contrast in distance between data points. Because of this, clustering algorithms measuring the similarity between pairs by distances over all dimensions of the data tend to break down in high-dimensional space as the number of dimensions increases [9, 16, 52, 79, 84]. In fact, the distance or neighborhood becomes less meaningful as the dimensionality of a dataset increases [2, 10, 33]. Based on the literature, I conclude that the relative distance of the farthest point and the nearest point converges to 0 for increasing dimensionality, d, as shown in Equation 2.

$$\lim_{d \to \infty} \frac{dist_{max} - dist_{min}}{dist_{min}} \to 0 \tag{2}$$

Cluster Optimization The second major problem with high-dimensional data is cluster optimization. Static global cluster optimization criteria mean the clusters obtained will be relatively uniform in shape, size, and density, even if these do not reflect the underlying relationships among the data [20, 55]. If the high-dimensional data are not in a metric space, then the standard distance measures aren't defined and can't be used for clustering [13, 22]. These two problems increase the difficulty of properly identifying natural clusters: clusters of arbitrary shape and size (see Section 1.1.1).

In practice, some points are correlated with respect to a given set of dimensions and others are correlated with respect to different sets dimensions. However, recent research shows that their intrinsic dimension is often much smaller than the dimension of the ambient space and the data points could be drawn down to the union of low dimensional subspaces [8]. Thus, such high-dimensional data lie close to the low dimensional structures corresponding to several classes or categories to which the data

belongs. Equally weighting all dimensions during clustering can introduce additional noise.

Algorithm Performance Bellman [9] points out that more dimensions results in more possible values and combinations of values. Due to this, performance suffers from the following issues.

- Like any optimization problem, clustering becomes increasingly difficult with an increasing number of variables.
- The discrimination between the nearest and the farthest points becomes poorer in high-dimensional data spaces.
- Since the clusters are defined by only some of the dimensions, the remaining irrelevant dimensions may interfere with finding the true number of clusters.
- In a dataset with many attributes, some attributes will be most likely correlated.
- The wrong selection of a similarity function used by a cluster analysis technique may lead to the discovery of some apparently similar groups at the expense of overshadowing other similar groups.

Extraneous Dimensions High-dimensional data may include many irrelevant dimensions. The relevance of certain dimensions may differ for different groups of objects within the same dataset. Since groups of data are defined by some of the available dimensions only, many irrelevant dimensions may interfere with the clustering algorithm to find clusters.

However, global dimension reduction methods may not yield combinations of dimensions that are most relevant to a particular region of the space (the "measure of locality" problem). If there are no (or not only) globally irrelevant dimensions, but

given sets of dimensions are irrelevant only with respect to certain sets of objects, then different clusters only exist in different subsets of dimensions.

Thus, the challenge of clustering is related to the problem of finding an appropriate subset of dimensions (subspace) to describe the similarity of objects belonging to the same group, and possibly different subspaces for different groups of objects [8,84]. The cluster objects reside in axis-parallel oriented, affine subspaces. Adding a fixed vector to the elements of a linear subspace of a vector space produces an affine subspace, or Euclidean subspace, of the complete data space.

Clustering methods that use all of the dimensions of a high-dimensional dataset are rare [8, 84]. If not all of the dimensions are relevant, clustering algorithms can be confused by the Euclidean distance metric and fail to detect true clusters. Similarly, noisy data or irrelevant dimensions can hide clusters. In fact, for a data point in the higher space, the number of data points in its neighborhood would be smaller than a data point in lower space due to the sparse distribution of the data points in higher subspaces.

Natural Clusters Based on the our experience with phenotypic complexity and interdependence, we expect to find natural clusters in many biological datasets. Finding natural clusters in high-dimensional data, where optimization is difficult and complete visualization is impossible, becomes extremely important.

A clustering algorithm returns homogenous groups in the input data according to a specific objective function or criterion. A natural cluster is a cluster of any shape, size, and density; and it can be non-disjoint (overlap in one or more dimensions) with any other cluster in the data. In any dataset, these characteristics can vary freely and be nonuniform over the clusters. Thus, objective functions that constrain these characteristics *a priori* will not find cohesive natural clusters.

Density-based subspace clustering algorithms do not restrict the shape and size

of clusters, and are able to obtain clusters of arbitrary shapes and sizes. However, the fixed density thresholds these algorithms use can miss natural clusters, and these algorithms assume the clusters are disjoint.

Thus, we need to design a clustering algorithm that organizes a collection of objects into subsets of non-disjoint clusters; and where the density of data objects is computed locally based on the data points in a proposed subspace. The question of finding generic natural clusters to characerize different phenotypes in mutant and inbred lines of maize is the focus of this dissertation.

1.1.2 Metric and Non-Metric Spaces

Informally, a metric space is a set of objects for which distances between all objects are considered togather. More formally, a metric space is represented as an ordered pair, $\big(\mathcal{M}, d\big)$, where $\mathcal{M}$ is a set and d is a metric function on M. A function $d : \mathcal{M} \times \mathcal{M} \rightarrow \mathcal{R} \geq 0$, for any $x, y, z \in \mathcal{M}$, must fulfill the following conditions:

- $d(x, y) \geq 0$: non-negative or positivity;
- $d(x, y) = 0$, if $x = y$: identity of indiscernible objects;
- $d(x, y) = d(y, x)$: symmetry; and
- $d(x, z) \leq d(x, y) + d(y, z)$: triangle inequality.

If all of these conditions are not satisfied, the space is non-metric. High-dimensional data will often form a non-metric space because the relative distances between any two points converge to zero with increasing dimensionality (Equation 2). Since distance measures become useless for discriminating between nearest and farthest points, I designed a clustering algorithm for high-dimensional data in *non-metric spaces*.

1.1.3 Complex Phenotypes as High-Dimensional Data

The observable and measurable traits of organisms are called phenotypes. Phenotypes and their component phenes are deeply interdependent and complex, so that changing one alters many others [57, 58]. Yield, disease resistance, growth, stress responses, and morphology are notable examples of these interdependent, complex phenotypes in maize. Complex phenotypes vary by interacting with each other, their genetic backgrounds, the environment, and management practices. Epistasis, pleiotropy, and variations in expressivity and penetrance further expand the range and characteristics of phenotypic features. All of these properties make complex phenotypes difficult to recognize, classify, and measure. But characterizing complex phenotypes is the essential first step to understanding and eventually manipulating them.

How can one use high-dimensional data to characterize and quantitate complex phenotypes? A complex phenotype is by definition a multivariate set of interdependent dimensions. Analyses that attempt to isolate one dimension from the others suppress and confound information about these interdependencies that are useful in determining how many distinct phenotypes one has. Discovering phenotypes that are shared among genetic, environmental, or management factors requires data that be standardized so that they are truly comparable across all the different factors on an experiment. Thus, characterizing complex phenotypes becomes a question of what clusters naturally appear in the HDD data.

Characterizing and clustering phenotypes accurately is essential to inferring the networks that cause complex phenotypes. Network inference remains a difficult problem, despite the efforts of many investigators [2, 15, 24–26, 29, 35, 48, 69]. A central difficulty of these methods is that they rely on bottom-up inference from molecular data. Our group is exploring an alternative, top-down approach that defines regions of the network by the natural clusters found in data forming the phenotypic space.

To do this, high throughput phenotyping, high-dimensional data derived from the phenotypes, and novel methods to discover the natural clusters of the data are required.

Our HDD data come from experiments in which the maize plants have been planted and grown in the field. In this experiment, the phenotype is lesions on maize leaves. To quantitate the lesions, the mutant leaves are photographed under standard conditions, the background is masked (Figure 1), each lesion is segmented (Figure 2), and the quantitative information on the lesions (Figure 3) is measured [46,47]. Every lesion has been represented as a set of high-dimensional vectors that form high-dimensional phenotypes. This high dimensional data has approximately 25 dimensions. These data objects may not form a metric space (this is hard to test) and the number of lesion objects per leaf is between ten to two thousand. Based on empirical observations, we hypothesize that this dataset's clusters are irregular in size, shape, and numbers of data points; and that the clusters are not mutually disjoint.

Figure 1: Masked Leaf Illustrating One Lesion Phenotype

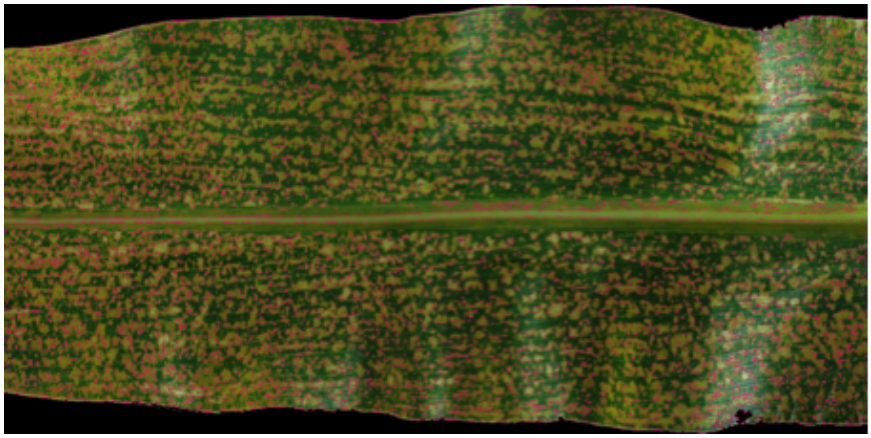


Figure 2: Lesioned Leaf after Segmentation [46]

Figure 3: Segmented Lesion

1.2 Overview of Clustering Approaches

In high-dimensional data many irrelevant dimensions can be expected. The relevance of certain dimensions may differ for different groups of objects within the same dataset. Since groups of data are defined by some of the available dimensions only, many irrelevant dimensions may interfere with the efforts to find these groups.

1.2.1 Dimension Reduction: Feature Selection and Transformation

Dimensionality reduction techniques have been broadly used to address this problem by reducing the number of dimensions. Moreover, dimensionality reduction techniques help to decrease time and space complexity, as well as to make data more comprehensible. The most popular dimension reduction techniques are feature selection and feature transformation. These remove irrelevant and redundant dimensions.

The existing methods of feature transformation and feature selection for dimension reduction are: PCA (Principal Component Analysis); SVD (Singular Value Decomposition); and wavelets or low frequency Fourier harmonics in conjunction with Parseval's theorem [4, 12, 16, 27, 42, 54, 55, 70, 86].

Dimension reduction methods map the whole feature space, $\mathcal{R}^n$, to a lower dimensional feature space, $\mathcal{R}^k$, where n and k are the number of dimensions in the feature space $\mathcal{R}(k < n)$. Feature transformation (PCA, SVD) and feature selection are very popular and traditional ways to reduce the dimensions. Feature transformation optimally transforms the original data space into lower dimensional space by forming dimensions that are linear combinations of given attributes. The new space has the property that distances between points remain approximately the same as before [16, 40, 44]. Correlation among attributes is often the basis for a dimension reduction in the feature selection technique. However, this technique has the major drawback in that it leads to a significant loss of information. These methods are not bijective in nature, so the transformed dimensions often have no intuitive meaning and so the resulting clusters are hard to elucidate. Because these methods are global dimension reduction techniques, they are not effective in identifying clusters in reduced dimension space.

These methods of global dimensionality reduction are inappropriate if the clusters lie in different subspaces of the dimension space. The subspace clustering algorithms evaluate dimensions only on a subset of the data, based on a "measure of locality" representing a cluster. These algorithms are able to uncover clusters that exist in multiple, possibly overlapping subspaces and represent them in easily interpretable and meaningful ways [4, 44].

1.2.2 Finding Subspaces

High-dimensional data spaces are inherently sparse [68]. It is possible and even highly probable that different subsets or combinations of dimensions (subspaces) may be relevant for different clusters. Thus, a global feature selection or dimensionality reduction method cannot be applied.

Another possible solution is finding subspaces, or subsets of dimensions, in the whole feature space where the natural clusters exist [4, 44, 53, 83]. The identification of subspaces by user is an error-prone process. Thus, two search strategies are implemented to find clusters in subspaces:

- *Top-Down approach*: This algorithm derives a cluster approximately based on the full-dimensional space, refines the cluster by adapting the corresponding subspace based on the current selection of points, and usually assigns each point to at most one subspace cluster [4, 12, 16, 27, 42, 54, 55, 70, 86].

- *Bottom-Up approach*: It starts with a single dimension and searches primarily for all interesting subspaces as combinations of lower dimensional interesting subspaces. It can assign one point to different clusters simultaneously. This approach tackles the problem of poor discrimination of the nearest and farthest neighbors [4, 12, 16, 27, 42, 54, 55, 70, 86].

But the probability of different subsets, or combinations of dimensions, may vary for different clusters, even for the subsets of highest probability; and selecting such high probability subsets is itself a difficult task because the search space of all possible subspaces of a d-dimensional data space is still $\mathcal{O}(2^d)$. Using local instead of global criteria can reduce the search space. Otherwise, an algorithm will not be able to unravel all the unseen natural clusters in data space.

1.2.3 Comparison of Existing Clustering Algorithms

Many algorithms to cluster data have been developed that use different strategies. When the data are high-dimensional, the size of the dataset is very large, the topological space formed by the data is uncertain, and the clusters are not uniform in shape, size, density, or other parameters of the objective function, not all of these algorithms will apply.

Partitioning Algorithms Partition-based clustering (k-means, k-medoids, and PAM) assumes the data form a metric space and uses a distance parameter to find spherical clusters. These algorithms are parametric in nature and are very sensitive to outliers. They expect that users will give the number of clusters (k) in advance, or experiment with different values of k to choose the "best" for their data. While these algorithms are very good for small and medium size datasets, they are less effective as the dimensionality of the data increase. They also can only detect spherical, disjoint clusters [56, 59].

Agglomerative Algorithms Agglomerative clustering methods (Single Linkage, Double Linkage, Average Linkage, and Ward's Method) do not require the expected number of clusters as input, and rely on comparisons of node similarity to terminate the computation. Similarity is measured by Euclidean distances between nodes and points: if the distance is small enough, points are merged into nodes and nodes into other nodes, ascending hierarchically. This agglomeration permits clusters of arbitrary shape, but constrains unmerged clusters to be disjoint. Apart from the reliance on a distance metric, the major drawbacks of these algorithms are their inability to correct erroneous merges or splits of nodes by backtracking; and the static order of point addition. Both these drawbacks limit their use with high-dimensional data [43].

Density-Based Algorithms One approach to finding natural clusters is to detect regions of the data space that differ in density. Regions of higher density, relative to their neighborhood, define clusters. The three main density based clustering algorithms are DBSCAN, OPTICS, and DENCLUE. They use a pre-set density threshold and terminate cluster formation when the density of the growing cluster drops below that threshold. These approaches handle outliers very easily. The major drawback of these algorithms is that they use a distance metric to restrict the density threshold values [5, 21, 32].

Grid-based clustering approaches circumvent the use of a distance parameter by dividing the data space into a multiresolution grid. Once cells with higher density are identified, the data space is partitioned into nonoverlapping grids. CLIQUE, ENCLUS, MAFIA, OptiGRID, STING, and WaveCluster are examples of this approach. These algorithms have fast processing time and the speed depends on grid size, not on number of objects. However, they have several limitations. First, grid sizes are determined using prior knowledge, rather than in an unsupervised fashion. Second, determining grid size is combinatorially explosive: as the dimensionality of the data increases, the number of combinations of grid sizes as 2^d. Finally, the clusters found are disjoint [4, 33, 34, 64, 76, 81].

Projected and Subspace Algorithms As the dimensionality of the data increases, data objects are unlikely to occupy all dimensions. In this situation, the ability to dynamically determine the combinations of dimensions that best characterize a cluster reduces noise in the clustering. Projection clustering, and its descendant subspace clustering, attempt to optimize the combinations of dimensions that best separate the clusters.

A classic example of projection clustering is P3C [16, 62, 72, 82, 85]. Projection methods partition the data into disjoint sets and discover correlations among data

objects in the hypothesized subspaces. Cluster quality is then evaluated using different objective functions, the choice being algorithm-dependent. This approach can find natural clusters. However, it scales poorly with dimensionality because identifying subspaces is combinatorially explosive. Under these conditions, the allocation of data objects to subspaces is ambiguous or incorrect.

Subspace clustering was developed to overcome these problems. These algorithms (such as DOC, PreDeCon, DISH, SUBCLUE, FIRES, CLIQUE, MAFIA, and ENCLUS, *etc*) define clusters as regions of relatively higher density in subspaces. They have different approaches to circumventing the combinatorial explosion of candidate subspaces (see below).

Subspace clustering algorithms can be divided into hard and soft subspace clustering. Hard subspace clustering uses heuristic criteria to find relevant dimensions. They search only a few combinations of these dimensions to improve speed, but the identification of the right subspaces is very hard. Soft subspace clustering algorithms determine the relevance of dimensions by choosing one or more seed points and weighting the dimensions in the neighborhood of those points. The weighting functions vary by algorithms [71].

For example, DOC measures density over a hypercube of fixed width to weight dimensions, rather like grid-based clustering. However, it is not able to find overlapping clusters and can't find subspaces larger than the hypercube's width [72]. PreDeCon, which is similar to DBSCAN, constructs a subspace preference vector based on the number of points that occupy each dimension. This vector is then used to assign objects to the subspaces. The major drawback of this algorithm is that as the dataset's dimensionality changes, it is unable to find the correct subspace clusters. It has better performance compared to DOC and PROCLUS [13].

DISH first computes the frequency of object values in each dimension, then groups

the dimensions by the similarity of their variances to find the subspaces and the objects they contain. *A priori* algorithms also use this "frequent item" approach. The assumption that dimensions in subspaces should have similar variance is difficult to justify, and seems very dataset-dependent. In practice, DISH performs poorly in high-dimensional spaces [1].

SUBCLUE measures distance around a seed point to define a density connected core object and its surrounding border objects [21]. In high dimensional space, it uses a gready approach to discover density connected clusters. The great feature of SUBCLUE is that it overcomes the limitation of grid-based clustering algorithms. It has better clustering quality in comparision to other subspace clustering methods like CLIQUE, MAFIA, ENCLUS, but runs more slowly. Another is that it has difficulties in identifying the core objects in different subspace cardinalities using same parameter settings.

Density-Based Subspace Clustering Density-based subspace clustering uses density-based clustering approaches to find subspaces and clusters. Density-based clustering algorithms can find arbitrarily shaped and adaptive sized clusters. The clusters have been identified through cluster density over some threshold; each point must have a minimum number of points within its "neighborhood". The density-based clustering algorithm principle makes it easy to find natural clusters. It also filters out outliers and handles noise in a better way in comparison to other clusters. This kind of algorithm is also computationally efficient because it needs only one scan and uses local cluster criterion, such as density threshold and density-connected points. A variant, scalable density-based subspace clustering, steers cluster mining to a few selected subspace clusters by identifying and clustering promising subspaces and their combinations directly [63]. By definition, any high-dimensional cluster must appear in many low dimensional projections. By mining only some of the low-dimensional

projections, the algorithm gathers enough information to jump directly to the more interesting high-dimensional subspace clusters without processing the other projections.

Density-based subspace clustering is very useful in finding subspace clusters in high-dimensional data. Since it scans data points only one time computationally, it performs better than other kinds of subspace clustering. It generates natural clusters of adaptive size, shape, densities and dimensionalities.

New Clustering Methods in Big Data Big data clustering [3, 61, 73, 77] is an exploratory technique. The choice of a clustering algorithm and its parameters is data dependent. The scalability, cluster accuracy and number of clusters are big challenges with heterogeneous data, streaming data, and validity. There are difficulties in applying clustering techniques to big data, because of the size of the data (terabytes and petabytes) and correspondingly high computational costs. Traditional clustering techniques cannot cope with this huge amount of data because of their high complexity and computational cost. For instance, the traditional k-means clustering is NP-hard, even when the number of clusters is two. Consequently, scalability is the main challenge for clustering big data [3, 61, 73, 77]. So the question is how to cope with these problems and how to deploy clustering techniques to big data and get results in a reasonable time [3, 61, 73, 77].

In general, big data clustering techniques [3, 61, 73, 77] can be classified into two major categories: single-machine clustering techniques and multiple-machine clustering techniques. Recently multiple machine clustering techniques have attracted more attention because they are more flexible in scalability and faster.

1.3 Problem Statement

Density-based subspace-clustering algorithms for high-dimensional data face many challenges. The significance of the local relevance among the data with respect to the subspaces (subset of dimensions) has led to the advent of the subspace clustering algorithm. Thus, different groups of points may be clustered in different subspaces, and a significant amount of research has been elaborated upon, which deals with subspace clustering, and aims at discovering clusters embedded in any subspace of the original feature space [16, 42].

As the search space for relevant subspaces for defining meaningful clusters is infinite, and uses the global density threshold in metric space, it is necessary to apply a novel approach using density threshold based on a "measure of locality" in "non-metric space". This makes the processing of identifying suitable subspaces containing clusters feasible [23]. Even the exponential growth in the number of these subspaces with the high-dimensionality of data makes the whole process of subspace clustering computationally very expensive [44].

Since data in a subspace are often distributed arbitrarily and not around a centroid, traditional clustering methods that take advantage of the spatial proximity of the data in each cluster are not in general applicable to subspace clustering of natural clusters.

There is need to apply density-based subspace–clustering approach on high dimensional data subspaces, to find natural cluster. With the varying region densities in different subspace cardinalities, we note that a more appropriate way to determine whether a region in a subspace should be identified as dense is by comparing its density with the region densities in that subspace. These subspace clusters may be partially or completely overlapping or non-overlapping to each other. These clusters contain redundant information which means that the data points are common in both

clusters C_i and C_j.

Due to the variation of density thresholds in different subspace cardinalities for discovering clusters, it is challenging for subspace clustering to simultaneously achieve high precision and recall for clusters in different subspace cardinalities. Thus, there is a need to propose a density-based subspace-clustering algorithm for finding natural clusters in non-metric space and that take into account the multi space structure of the data as well. An algorithm should use locally adaptive density threshold for each subspace in non-metric space instead of global density threshold in metric space. Therefore, we proposed a novel density-based subspace-clustering algorithm called DynaDASC (Dynamic Locally Density Adaptive Scalable Subspace Clustering). This algorithm uses a relatively local density adaptive density threshold value. It is scalable, dynamic and nonmetric in nature as well.

1.4 Contributions

In this dissertation we contribute a solution of three major tasks in data analytics for maize phenotype. Firstly, we proposed novel density-based subspace clustering algorithm, called DynaDASC, to account for challenges posed by high-dimensional data. Secondly, we want to verify the interdependence of the phenotypes are interdependent in our sample data. Finally, we want to contribute non-parametric standardization technique, called SDFS (Standardization for Distribution Free Statistics), to preprocess the low dimensional and sparse dataset.

2 Materials and Methods

2.1 Datasets

Two experimental datasets have been used in this dissertation. These experiments are conducted in a greenhouse and outside in a field for maize (*Zea mays*) phenotypes. In both experiments, we are interested in determining how many distinct phenotypes exist and determining the relationships among phenotypic dimensions and input variables such as stress condition, inbred background, and genotype.

2.1.1 Greenhouse Experiment with Combined Stresses

In experiment one, conducted in Dr. Ann Stapleton's laboratory, ninety different inbred lines of maize were planted and grown in a greenhouse under the stress of nine different combinations of water and nitrogen fertilizer. These growth conditions are shown in Figure 4.

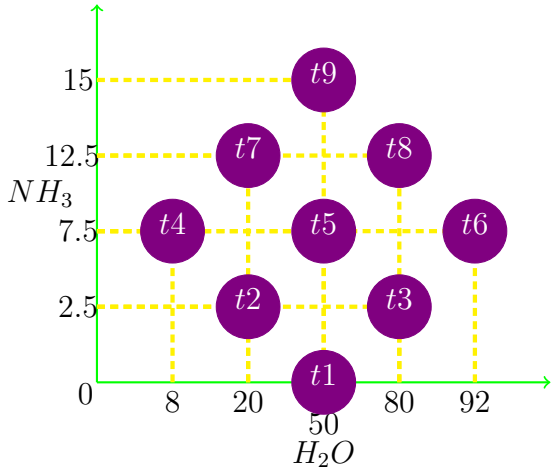


Figure 4: Growth Conditions

It is of biological and genetical interest to determine how many different types of response (phenotypes) exist, independent of the inbred lines and the stress combina-

tion.

Forty plants of each line were grown; four plants under each growth condition, except for growth condition five, which had eight plants. The experimental design was optimized to detect nonlinear interactions between the stresses. Many plants died during the experiment, so the number of surviving plants is often much lower. The distribution of the survivors over all the subpopulations is shown in Figure 5.

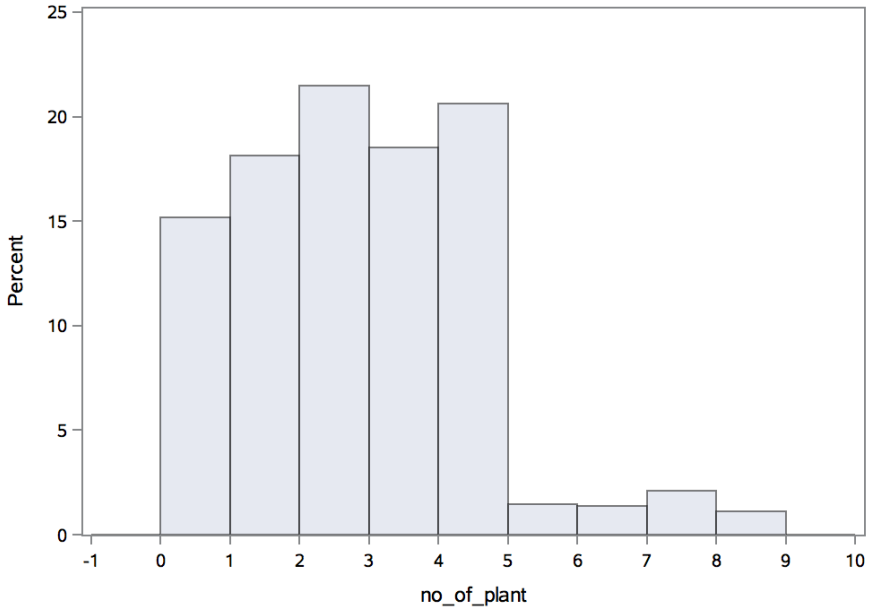


Figure 5: Distribution of Surviving Plants over all Strata. The number of plants in each bin is the value at the left boundary.

Eight types of paired phenotypic information were collected before and after the stresses were applied. The phenotypes plant height, canopy spread and stem diameter were measured before and after applying the different growth conditions. Not all values for all plants were measured, making the data sparse; and the different inbred lines exhibited different ranges of data values. The small sample sizes make it impossible to determine which parametric distribution best fits the data, so they were assumed to be non-parametric.

The data distribution and frequency of the original and rescaled data along three

dimensions — $\Delta h, \Delta c,$ andΔs — are shown in Figure 6. The histograms of the original Figure 6(a) and rescaled Figure 6(b) data show these dimensions are asymmetric and have mixtures of positive, negative and zero data values. They also illustrate how wide the range, shape and central location of the data are. The distributions of the three dimensions and their deviations from normality in the normal quantile-quantile plots indicate the dimensions are not normally distributed, consistent with our assumption that the data are nonparametric. Moreover, three dimensional plots of the original and rescaled data, shown in Figures 6 (a) and (b), show that the data points are so compact and overlapping each other that they cannot be visually separated. The rescaling was by min-max normalization between 0 and 1.

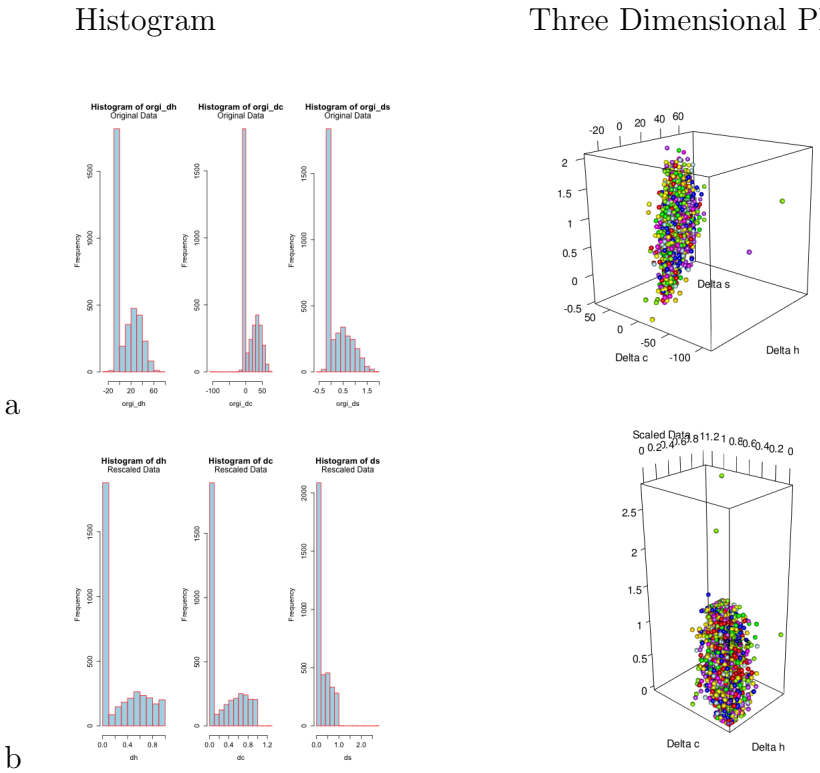


Figure 6: Distribution of (a) Original, and (b) Rescaled Data

Therefore, we standardized this dataset to unravel the information hidden in it.

2.1.2 Field Experiment on Lesion Mimic Mutants

The disease lesion mimic mutants of maize exhibit a very wide range of phenotypes that are sensitive to the plant's genetic background and environmental perturbations [39, 65, 66]. Variations in lesion morphology alone encompass at least ten different dimensions (areas, shape characteristics, border characteristics, lesion color, internal morphologies, ground tissue colors, local lesion density and position, and distribution on the leaf). These variations make lesion phenotypes an ideal high-dimensional, complex phenotype to explore clustering methods.

In experiment two, conducted in our laboratory, the plants are grown in the field, either in Columbia in the summer field seasons or on Moloka'i, Hawai'i in winter nursery (Figure 7). To quantitate the lesions, the mutant leaves are photographed under standard conditions, the background is masked, each lesion is segmented, and the quantitative information on the lesions is measured. Every lesion has been represented as a set of high-dimensional vectors that form high-dimensional phenotypes.

Fourteen different disease lesion mimic mutants were back crossed into three different inbred maize lines. The number of back-crosses varied among the different line/mutant combinations. Leaves were photographed shortly after the mutant plants had finished shedding pollen. At this stage, some combinations had lesions that had reached stasis and had well defined boundaries, while other combinations had lesions that continued to expand and differentiate. In some cases, this latter class of phenotypes produce a diffuse chlorotic zone surrounding one or more necrotic central lesions. As the chlorotic zones expand, intensify, and merge, it becomes difficult for humans to identify the original lesions [45].

Figure 7: Experimental Field

2.1.3 Low-Dimensional, Sparse Synthetic Data

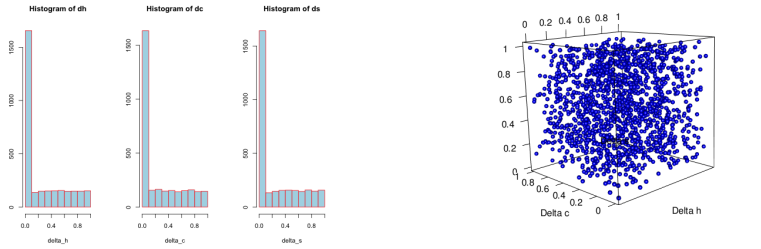
To compare the effects of the standardization methods, we needed a sparse dataset that had none of the internal relationships one would expect of real data. A three-dimensional random uniform distribution by definition has no internal structure, so we began with that and then salted the distribution with zeroes at random positions to make the data sparse. Sparse, non-parametric data are common in many biological experiments, such as the greenhouse experiment of Section 2.1.1.

Four sparse three-dimensional matrices of 3000 (47% zeroes), 6000 (52% zeroes), 7000 (55% zeroes), and 10,000 (53% zeroes) random, uniformly distributed points were generated using methods in Python's `numpy` and `scipy` libraries. Matrices of the appropriate size were filled with pseudo-random numbers in the $[0, 1]$ interval using the Python library. The first n rows corresponding to the final percentage of zeroes in the matrices were set to zero, and then the row and column indices of the matrices were shuffled to randomize the positions of the zeroes throughout the

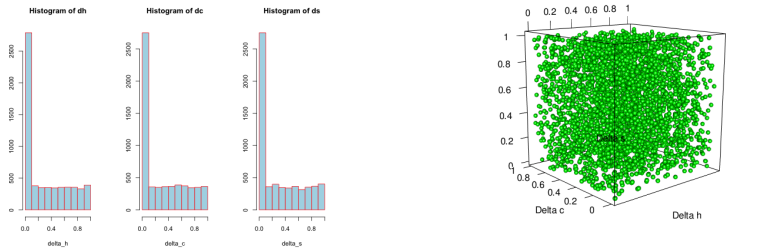
matrices. The resulting matrices combined a sharp peak at zero with an essentially uniform distribution. Figure 8 shows the histograms, and the three-dimensional plots, for each synthetic dataset. No internal structure is visible in any of the datasets.

Histogram Three-dimensional plot

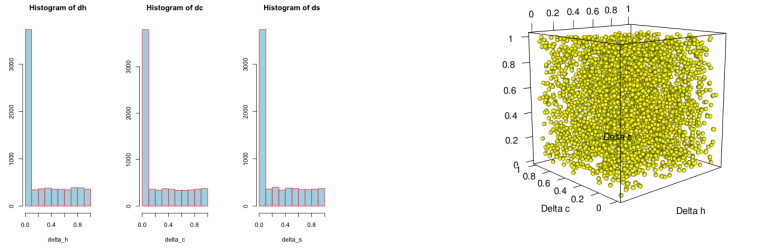
a



b



c



d

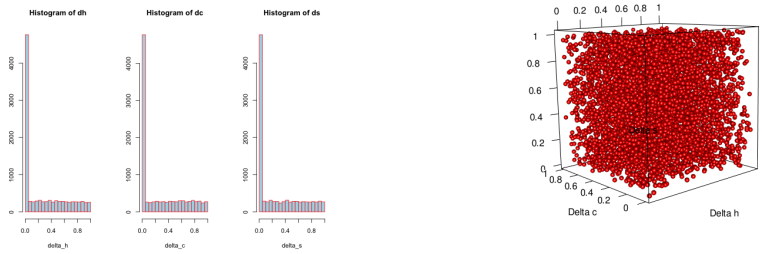


Figure 8: Univariate Distributions of Synthetic Data. (a) Size: 3000, (b) Size: 6000, (c) Size: 7000, and (d) Size: 10000

2.1.4 High-Dimensional Synthetic Data

The DynaDASC alogrithm was tested using a synthetic dataset generated by Dr. Kazic. A producing function describing the relative changes in plant height, z, in response to combined drought (x_w) and nitrogen stresses (x_n) is shown in Equation 3:

$$z = c\left(ax_w^2 + bx_n^2\right) + dx_w + ex_n \tag{3}$$

This nonlinear model qualitatively accounts for the behavior of the maize lines in the greenhouse experiment of Section 2.1.1 (Chang *et al.*, in preparation). The model was tested by simulation, varying the values of the parameters a, b, c, d, and e over combinations of parameter values. A total of 34 proxy functions that characterize different features of the resulting three-dimensional surfaces were computed. Values for the nine most informative proxies formed the test data points for clustering. The dimensions used were:

- z_max, the height of the surface's peak;
- *row*, the position of the peak along the water axis;
- *col*, the position of the peak along the nitrogen axis;
- the absolute value of the volume of the surface between its maximum and the minimum of the Mo17 surface, which is the lowest point among the experimentally determined surfaces;
- *level*_0.75, the number of matrix cells within 75% of the B73 peak, the highest experimentally determined surface, ±0.01;
- *level*_0.50, similarly for 50%;
- *level*_0.25, similarly for 25%;

- *curvature_max*, the discrete curvature in the immediate neighborhood of the peak [80] ; and
- *area_max*, the area in the immediate neighborhood of the peak.

The data values have been selected for DynaDASC based on the data ranges. We thresholded at different data point values for different dimension based on histogram output observation. It has been considered that we should cover large range of data objects and histogram bins sholud be spreaded along the axis such that we could see their ranges very easily.

2.2 Methods

2.2.1 Data Transformations

Normalization of Greenhouse Data Our first step for the experimental data was to rescale the range of each dimension using the min-max normalization. For each of i dimensions, the rescaled value, $v_{r,i,j}$, of that dimension for every original point $v_{i,j}$ is

$$v_{r,i,j} = \frac{(v_{i,j} - min(i,\cdot))}{max(i,\cdot) - min(i,\cdot)},$$

where the $\cdot$ denotes the value of all the points $v_{i,\cdot}$ in dimension i.

Standardization for Distribution-Free Statistics (SDFS) SDFS sets $g(\mathbf{x})$ to be the minimum of each dimension for each biological class, treating the trivariate dimensions as independent. γ is simply the measured values of the variates in each class. Thus,

$$f_s(\mathbf{x}) = \frac{f(\mathbf{x}) - min(f(\mathbf{x}))}{f(\mathbf{x})},$$

where $f_s(\mathbf{x})$ represents the value of standardized dimensions, and $f(\mathbf{x})$ and $min(f(\mathbf{x}))$ represent the measured dimensions and the minimum value of measured dimensions

across groups, respectively.

Data Standardization by Parametric Methods We tested many parametric approaches to standardization, comparing the clusters obtained for the low-dimensional experimental biological data to those produced by MODECLUS on SDFS-standardized data. The methods included L, mean, median, STD, AGK, Euclen, AHUBER, AWAVE, IQR, MAD, Maxabs, USTD, MidRange, Spacing, and Range [7, 11, 28, 28, 36, 37, 41, 67]. Mid Range gave clusters most comparable to those produced by SDFS in both number and position in the three data dimensional space, while ABW gave very few clusters collapsed in one corner of the space.

Orthonormal Transformation of Greenhouse Data Quantitative geneticists commonly treat complex phenotypes as if each phenotypic component was linearly independent of the others. To simulate this situation, we transformed the experimental data of Section 2.1.1 by finding an orthonormal basis for them. An orthonormal basis forces the dimensions to be linearly independent; thus, we synthesized three "independent" univariate dimensions. We used the *orth* function of R's `pracma` package to find the orthonormal bases [14]. The orthonormally transformed data were used in the clustering experiments of Section 3.1.4.

2.2.2 Acquisition of High-Dimensional Lesion Data

The field experiment includes construction of near-isogenic lines of lesion mimic mutants in three inbred backgrounds; acquisition of phenotypic and genetic data and tissue; image processing; and data pre-processing. The result is a high-dimensional dataset with changes in genotypes and phenotypes tracked throughout the pedigrees of the lines.

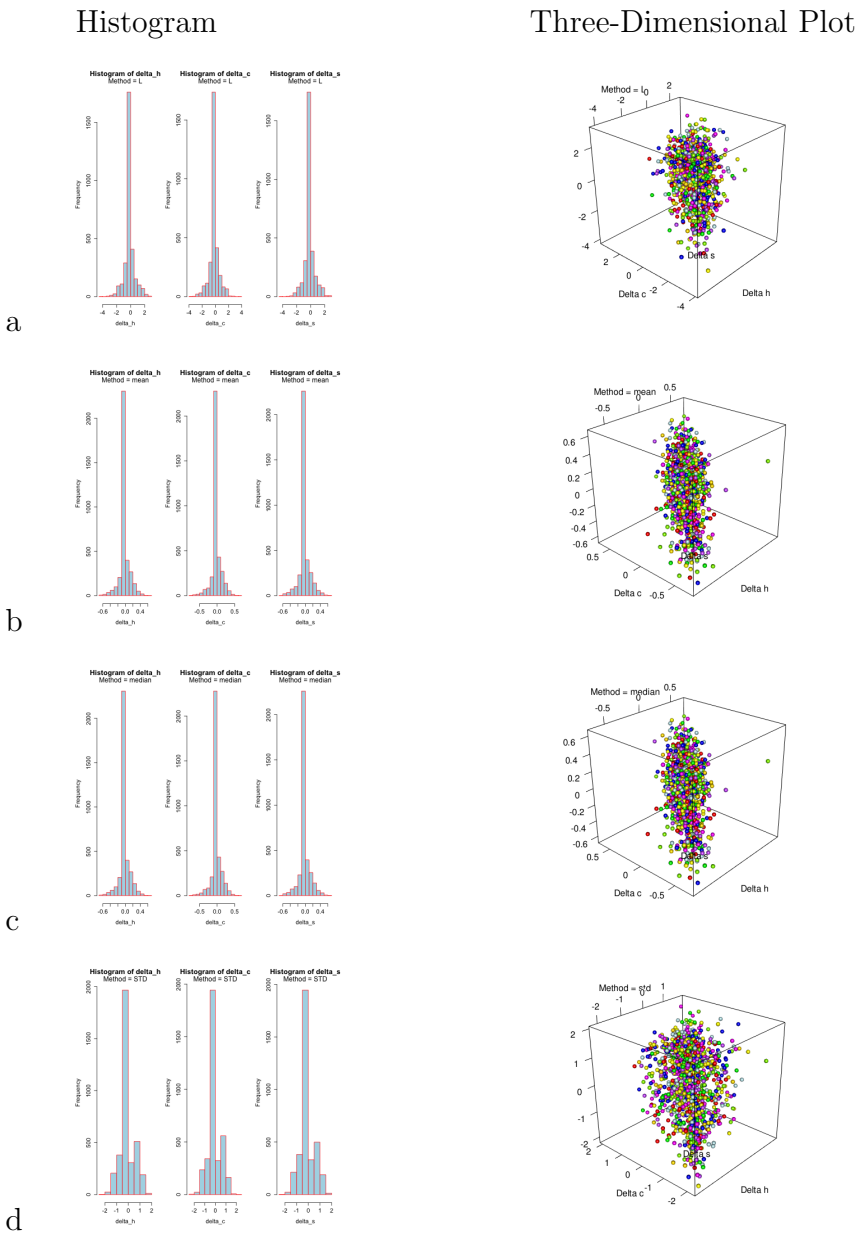


Figure 9: Distribution of Standardized Data (a) L, (b) Mean, (c) Median, and (d) STD

Line Construction and Management of Seed and the Field The field experiment starts with the selection of seed and its packing. Mutants are back-crossed six times to their recurrent inbred parent to reduce phenotypic noise to a reasonable level. Selection of offspring for the next step in line construction maximizes the

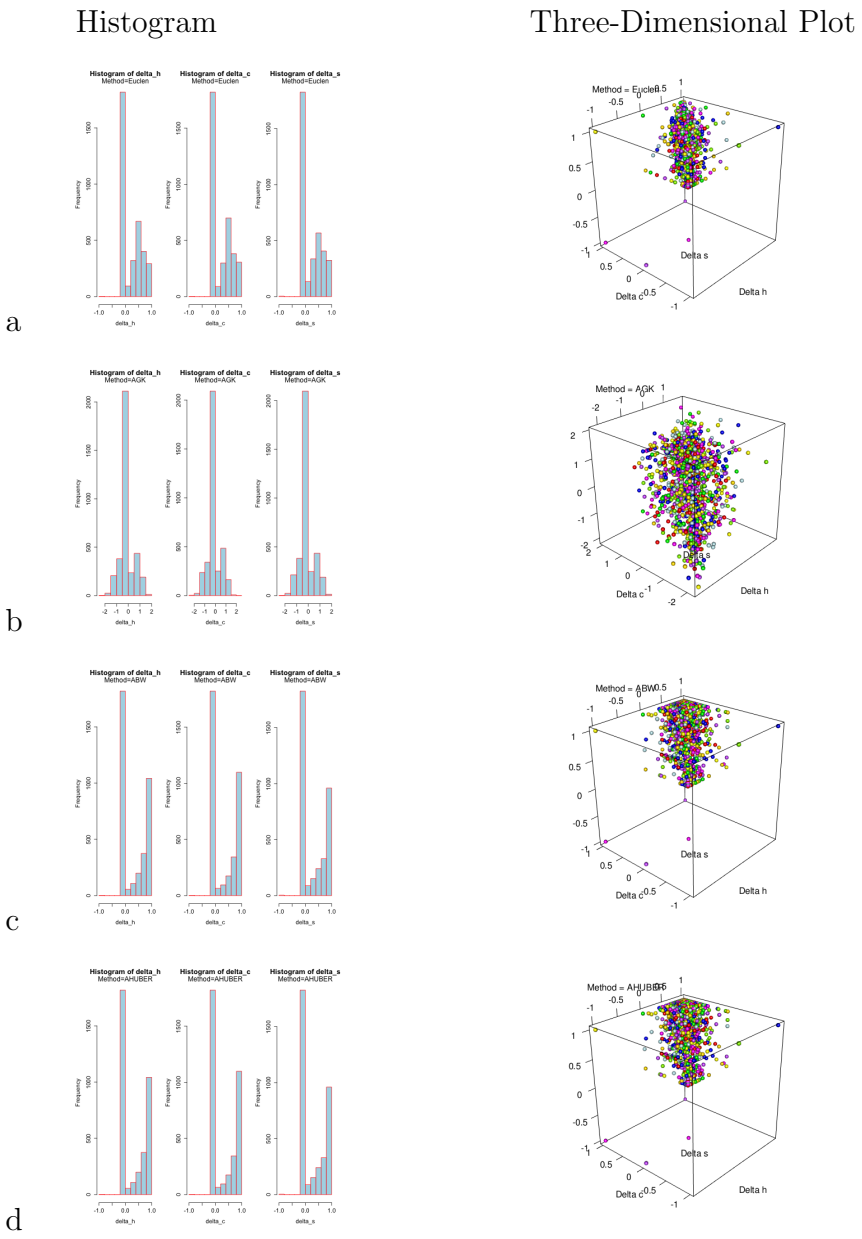


Figure 10: Distribution of Standardized Data (a) Euclen, (b) AGK, (c) ABW, and (d) Ahuber

clarity of phenotypic expression and the existence of images of parental phenotypes, and attempts to optimize the overall vigor and reproductive status of the mutant male parent. Many lesion mimic mutations produce collateral metabolic effects that depress vigor and reduce or eliminate ear production.

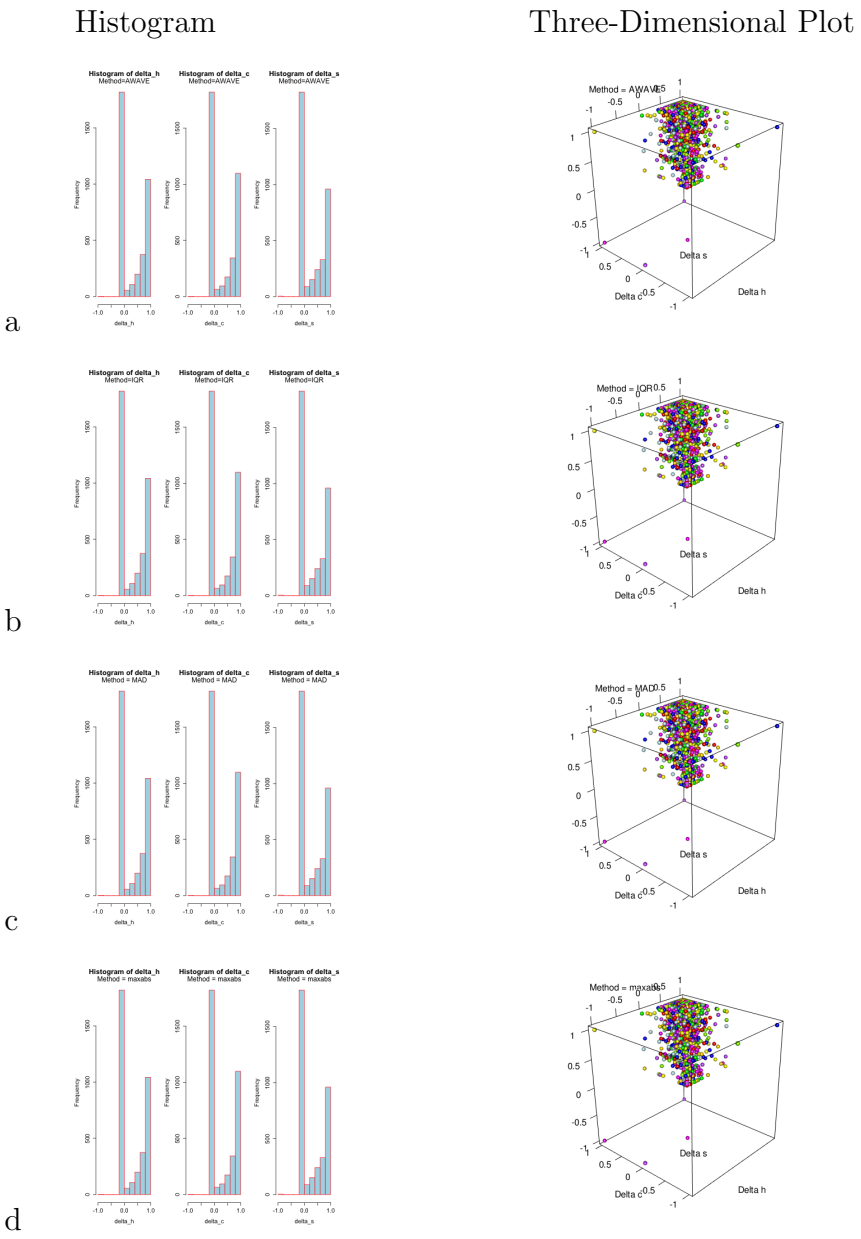


Figure 11: Distribution of Standardized Data (a) AWAVE, (b) IQR, (c) MAD, and (d) Maxabs

The seed packets (Figure 14) are labeled with the packet number, numerical genotypes of the parents, row number (to be planted in the field), planting number, number of seeds, length of row, and sleeve number where seed from prior crops is stored in the seed cold storage room of the Sears greenhouse at University of Missouri. The

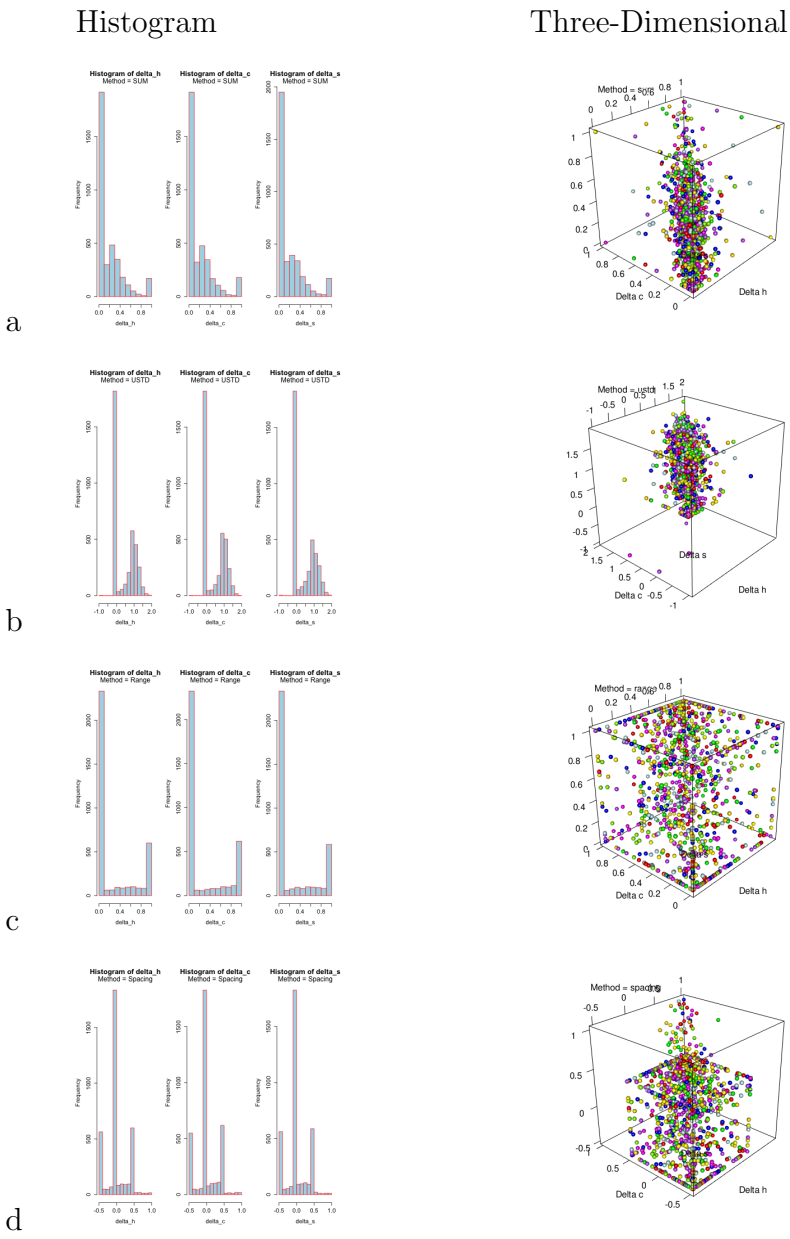
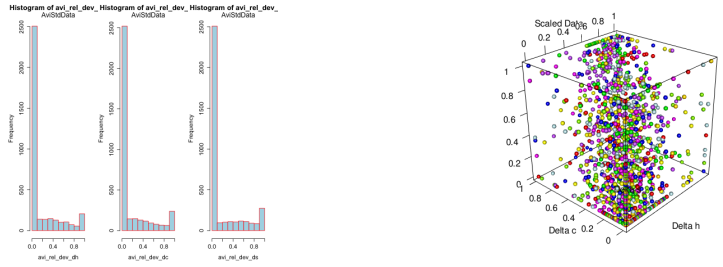


Figure 12: Distribution of Standardized Data (a) SUM, (b) USTD, (c) Range, and (d) Spacing

numerical genotype of a plant is unique and is the primary key in the database for all objects derived from that plant. It includes the crop number, any inbred identifier, row, and plant number. Offspring are identified by the numerical genotype of the maternal plant, since only one ear is ever used from a plant. The aim is to produce

Histogram Three-Dimensional Plot

a



b

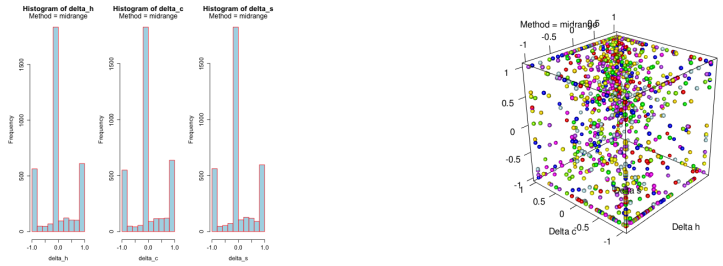


Figure 13: Distribution of Standardized Data (a) SDFS, and (b) Midrange

a complete provenance trail for all biological materials and the data derived from them [45].

Figure 14: Seed Packet (left) and Box (right)

Rows are labelled with barcoded row stakes; each stake and seed packet are scanned immediately before planting and the planting timestamped. The corn is

planted in either full 20' rows or half 10' rows at varying densities, depending on whether the line is an inbred intended for seed production (12" spacing); a vigorous mutant for observation and pollen production (8" spacing); or a mutant with poor germination (4–6" spacing, since many mutant offspring will die very early). 4' alleys separate each block of rows. Border rows of sweet corn surround the experimental corn to help decoy predators. Seed is planted manually using a jab planter. Delayed plantings of inbred and mutant lines are used to accommodate the faster growth rates of less back-crossed lines. Figure 15 illustrate the field and planting the corn seeds.

Figure 15: View of Field Tending and Planting

Once all rows are fully germinated, we count the plants in each row ("stand counts"), noting the approximate growth stage and any observable phenotypes. We then generate a tag for each plant that includes multiple copies of its numerical genotype as both text and barcode (Figure 16).

Figure 16: Bar Code

To speed field operations by minimizing look-ups of important information, the tag also includes an abbreviated symbolic genotype of the parents, family number, and a tracking number for the pedigree branch. These bar codes are used for identification of the plant, its leaves during photography and tissue sampling, its pollen or ears during controlled pollinations, and keys during collection of other types of data, such as mutant phenotype, plant height, and narrative descriptions [45]. Tag data are printed on perforated card stock, laying out the tags vertically through a stack of sheets. The block of sheets is drilled at one end, one hole for each stack of tags, sawn apart with a bandsaw, and threaded onto a pin made of galvanized #9 wire. This keeps the tags in the correct orientation and order in a set of packets so that they can be rapidly stapled around each plant.

To mimimize any effects of mutant cytoplasm, controlled pollinations are always performed using the mutant plant as the male parent. Ear shoots are covered with a waxed bag before the silks emerge to protect them from ambient pollen, then cut back the day before to produce a vigorous flush of new, receptive silks. Tassels of selected plants are covered with paper bags to collect the pollen before it sheds on the morning of pollination. Fragile tassels, or tassels that will be used for large numbers of pollinations, are covered briefly with a glassine bag for pollen collection. Working swiftly, the waxed bag is removed, the pollen dumped on the fresh silks, and the pollinated ear covered with the paper tassel bag to prevent contamination. Tear-off tags from the maternal and paternal parents are stapled to the bag to identify the cross. The pollination is recorded immediately by scanning the barcodes, and the data are timestamped so we can know what was done and debug unsuccessful efforts.

Pollinated ears are allowed to mature for at least 40 days before harvest, cleaning, and drying. The dried corn is shelled, the kernel count estimated, any notes written on the bags are recorded in the database. The maternal and paternal tags from the bag are stapled together to identify the seed, and this label is then stapled to the bag containing the seed. The corn is then filed in boxes for cold storage.

Acquisition of Data and Tissue Our goal is to image a leaf expressing lesion phenotypes from every plant used as a male during pollinations, and other mutant plants as needed for comparison. Lesion-bearing leaves are identified and photographed *in situ* or *ex situ*. In both cases, the leaves are placed on a field of dark blue cloth that includes an X-Rite Mini Color Checker Classic and the barcoded tag from the source plant. The relative number of the leaf is written manually on the tag. The Color Checker is included to permit color correction among images taken under varying light conditions and provides an internal size standard. Thus, every image is self-identifying and includes internal photographic standards [45, 46].

Images were taken with a Nikon D80 10.2 MP DSLR camera with an AF MICRO NIKKOR 60 mm lens. Using purpose-built jigs, leaves are held parallel to the plane of the lens. For *ex situ* images, leaves are cut from the plant, a tag stapled to them, and the cut end is immersed in ice water. The leaf is rinsed, air-dried, and photographed within an hour. *In situ* leaves are sprayed with water, gently wiped, and air-dried before imaging. The *ex situ* apparatus holds the leaves at a fixed distance and illuminates them along their length and from above and below with fluorescent 5000 K lamps. Light is also reflected by aluminized bubble wrap enclosing the back, top, bottom, and part of the front of the apparatus, to further diffuse the angles of incident light. Figure 17 demonstrates our setup for image capture.

Most images included a third or more of the maize leaf; smaller leaves entirely filled the image [46]. *In situ* images are photographed with shading to reduce reflections

Figure 17: Jig used for Image Capture. Views from front (left) and side (right)

from surface waxes. A skilled team of three people can image approximately 300 leaves/day *ex situ*; two people can manage about 50 leaves in a morning *in situ*. These photographs are used to extract phenotypic data with the help of MATLAB's digital image processing toolbox, described in Section 2.2.2.

In addition to imaging phenotypes, we collect phenotypic information on each plant in the mutant families (wild-type or lesion mimic mutant, relative or absolute plant height, quality of reproductive organs, and other phenotypes). Leaf tissue is collected from imaged plants, either immediately after *ex situ* photography or during data collection in the field for *in situ* imaged plants. A strip of leaf is inserted into a shoot bag punctured in the center, an identifying sample tag stapled on the closed bag, and the sample tag and plant tag scanned. Tissue samples are lyophilized until dry and then stored at -20° C pending extraction of DNA and whole-genome sequencing. Narrative descriptions of the field, mutant families, and other information are recorded throughout the field season.

Image Processing We used DCRAW, a raw decoding open-source ANSI C program, to convert the raw image file format of the camera (.NEF) to .tiff [17]. The

original photographed leaf, shown in Figure 18, consists of blue image background,

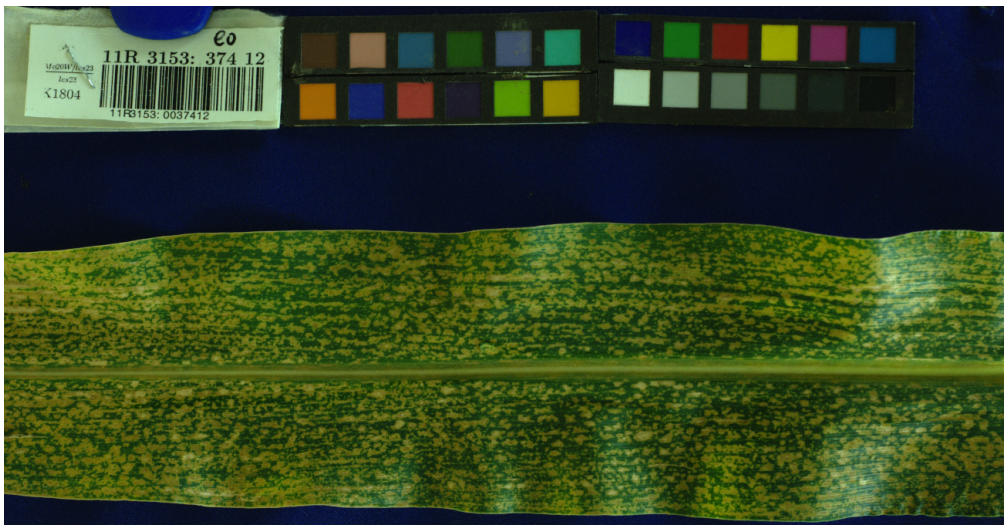


Figure 18: Photographed Leaf

bar tag, and X-Rite Mini Color Checker Chart. To segment lesions from leaf image we will need to remove all other components on image except the leaf and make its background black by masking. We use `regionprop` (a MATLAB function) and color based thresholding to segment these four components and save them separately for further use in color calibration of the image. Among these four components, one of them is masked leaf shown in Figure 19.

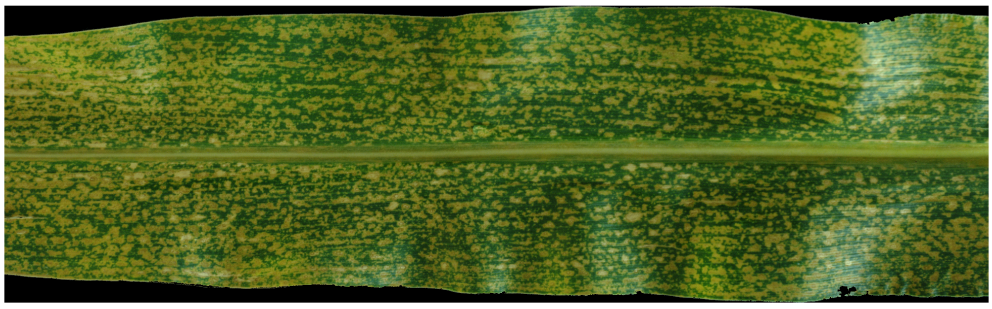


Figure 19: Masked Leaf

We then use our segmentation cascade to segment the lesions from the background

tissue [46]. Segmentation uses a cascade of three algorithms. These three algorithms are multiresolution analysis for lesion detection, gradient vector diffusion to find approximate position of lesions, and active contours to refine lesion boundaries. The segmented leaf is shown in Figure 20.

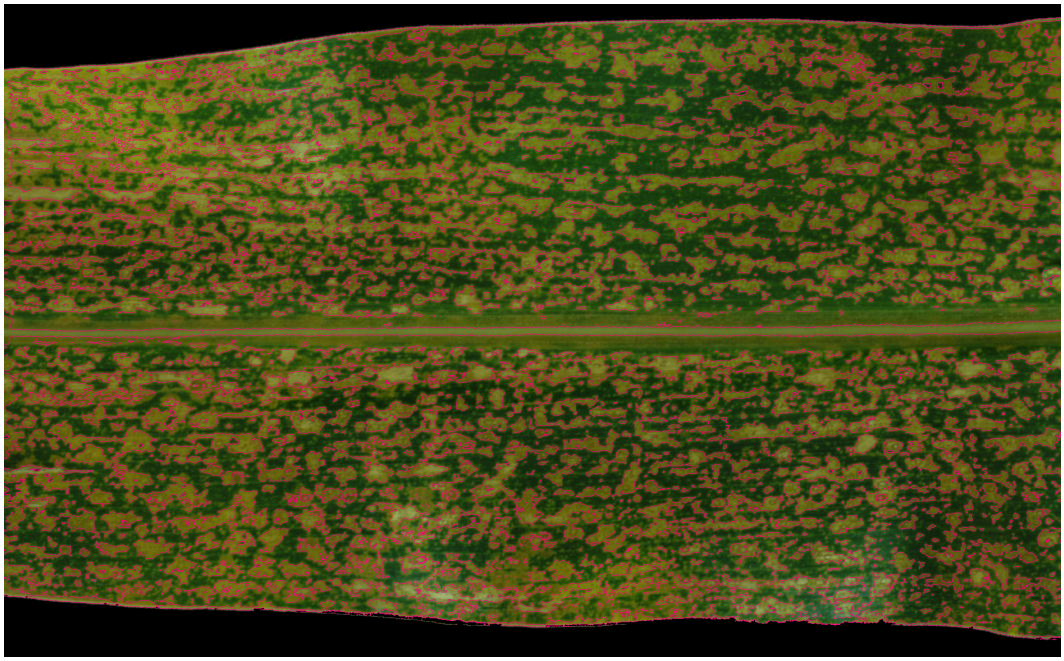


Figure 20: Segmented Lesions

We performed a preliminary characterization of the phenotypic components of the segmented lesions. These phenotypic components are the lesions area, perimeter, major axis length, minor axis length, eccentricity, convex area, convex hull, centroid, filled area, filled Image, Image, equivalent diameter, Euler number, extent, extrema, orientation, pixel list, solidity, subarray index, red color mean, blue color mean, and green color mean. So each lesions phenotypes are described by high-dimensional data. Single leaf consists of 2000 – 3000 (approximately) number of lesions. The data values of phenotypes are currently saved in xlsx spread sheet shown in Figure 21.

Per-Lesion Data						
Lesion Number	Area	Perimeter	Major Axis Length	.	.	.
1	605	167.4386	49.1983	.	.	.
2	120	46.0416	15.948	.	.	.
3	336	84.669	26.1619	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.

Figure 21: High-dimensional Dataset

Pre-processing of High-Dimensional Data In order to find correct clusters in a given dataset, data preprocessing is required. The first step of preprocessing is min-max normalization of dimensions to make it comparable to each other. This normalization technique is described in detail in Section 2.2.1. The second step of data preprocessing is standardization to equalize the variances of the dimensions so that data object assigned to the correct clusters or subspaces. The given high-dimensional dataset needs parametric analysis. Thus, parametric methods of standardization technique has been used. The standardization techniques are described in Section 2.2.1. The third step to find the correlation among dimensions. If dimensions are closely correlated to each other then we use dimension reduction techniques. The dimension reduction techniques are described in Section 1.2.1. The correlation matrix plot was used to show the bivariate correlation among dimensions.

2.2.3 Clustering Methods

MODECLUS MODECLUS is a density-based clustering method which performs cluster analysis for non-parametric data [74]. Since it is a density-based clustering algorithm, it can find clusters having irregular shapes and different sizes. MODECLUS starts with n or fewer seed points and describes an enclosing sphere of

radius r around them, where r is a variable specified by the user. It computes the local maxima of the probability density function inside the sphere to identify regions of greater density, defined as $n_i/(nv_i)$, where n_i, n and v_i denote the number of neighbours, the sample size and the volume of the neighbourhood at point x_i respectively. It then attempts to hierarchically merge these dense regions, based on the significance test. That test depends on the smoothing parameter, which is also specified by the user. This smoothing parameter is not only the basis for the statistical significance test, but also determines the number of clusters present in the dataset. Since the significance test helps in computing the number of clusters, users never need to specify the number of clusters. They only need to specify the smoothing parameter and the radius r. We used method six of MODECLUS in this experiment because this method begins with all observations unassigned [49, 50].

We tested the effects of the different standardization methods using the non-parametric density-based MODECLUS procedure of SAS. MODECLUS can give clusters of arbitrary shape without constraining their number [31, 49, 50, 74]. However, the algorithm is restricted to relatively low-dimensional data. The values for the MODECLUS parameters radius and threshold were tuned until the number and stability of the clusters for the experimental data were optimized.

Other Clustering Methods The DynaDASC clustering algorithm has been compared with a diverse set of existing algorithms. These algorithms are hierarchical clustering (Single Linkage, Complete Linkage, and Ward Method), partition based clustering (K-Means, K-Medoids), model based clustering, density-based clustering (DBSCAN, OPTICS), projected clustering, subspace clustering (CLIQUE, FIRES, SUBCLU). The R packages `DBSCAN`, `subspace`, and `orclus` have been used to compute the results of these clustering methods.

Pseudocode of DynaDASC The pseudocode of generic functions and their steps for DynaDASC are specified in Figure 22, Figure 23, Figure 24, and Figure 25. The procedure – 1 (shown in Figure 22) computes relative adaptive density threshold (τ_k) value of data object and assign it to respective subspaces based on density threshold (θ_i). These subspaces are passed as an input argument to Procedure – 2 (shown in Figure 23). It optimizes subspaces using "principle of maximum entropy" of Graham Wallis theorem based on entropy. It also computes the skew factor and kurtosis factor of subspaces. Thereafter, these three values (entropy, skew factor and kurtosis factor) are visualized through two dimensional plot and decided the threshold values of $entropy_\theta$, $skew_\theta$, and $kurtosis_\theta$. These threshold values are used to optimize subspaces and saved it as optimized subspaces. The Procedure – 3 (Figure 24) takes optimized subspace as input argument and compute the number and name of dimensions of dataset present in subspaces. If number and name of dimensions are same then the subspaces are merged togather, othewise they are considered as seprate subspace. Thereafter, the Linderberg–Feller Central Limit Theorem or Lyapunov conditions applied on all merged and unmerged subspaces and check the convergence of all subspaces individually. If they are converged then it implies that all dimensions available in subspaces are comfortable. Finally, Procedure – 4 (Figure 25) computes correlation values of these subspaces and saved it for future use for computation of density threshold in Procedure – 1.

How Goodness of Clusters was Evaluated The goodness of clusters have been evaluated by computation of entropy and F-measure. The computation of entropy, ($E(C)$), for clusters is another way to evaluate clusters quality. Lower value of entropy implies that better is clustering. It can be measured by following formula.

$$E(C) = -\Sigma_{c_j}(\frac{n_j}{n})\Sigma_i(p_{ij} * log(p_{ij}))$$

Algorithm 1 DynaDASC algorithm

1: **procedure** 1. RELATIVE_ADAPTIVE_DENSITY_THRESHOLD(Dataset: std_dataset, Dimensionality: no_dimension, Number: data_objects)
2: Output: $SSID_k$ *and* $subspaces_k$
3: **FOR** $i \in \{1, \ldots, data\_objects\}$
4: **FOR** $j \in \{1, \ldots, no\_dimension\}$
5: compute probability of each $data\_point_{i,j}$
6: compute information contents of $data\_point_{i,j}$
7: find relative adaptive density of $data\_point_{i,j}$ using

$$\tau_k = (\lambda_{i,j} * \frac{N}{\delta^k}) + \Psi_k + (h_k * \Delta\tau_k)$$

8: **If** $(tau_k \geq th_1)$ & $(tau_k < th_2)$
9: assign $datapoint_{i,j}$ to $SSID_k$ and save $subspace_k$ values
10: **EndIf**
11: **ENDFOR**
12: **ENDFOR**
13: **return**($SSID_k$ index values of data points and unique $subspaces_k$)

Figure 22: Computation of Relative Adaptive Density Threshold and Subspace Allocation

where

$$p_{ij} = \frac{n_{ij}}{n}$$

, C_j is j^{th} cluster in C, n_j is number of data points in C_j that actually belong to subspace SSID.

The F-measure [38] is used to measure computing performance of different clustering algorithms. It is also known as evaluation function and described as follows.

$$F = \sum_{j=1}^{n} (\frac{n_j}{n}) * max_{(1 \leq k \leq \mathcal{K})} \Big(\frac{2n_{jk}}{(n_j + n_k)} \Big)$$

where n is total number of objects, n_j is number of objects in class j, n_k is number of objects in cluster k, and n_{jk} is number of objects occurring in both class j and class k, and k is the number of clusters equal to the number of classes.

1: **procedure 2.** SUBSPACE_SSIDK_OPTIMIZATION(SSID_k.csv)
2: Output: *SSID_k.csv file, leftover_SSID_k.csv*
3: **While** *not(SSID ∈ max (k))* **do begin**
4: apply "Principle of Maximum Entropy" using Graham Wallis
5: compute

$$p_k = \frac{n_k}{N}$$

where p_k is k^{th} subspace
6: compute value of W

$$W = \frac{N!}{(n_1! * n_2! * \ldots n_k!)}$$

7: check convergence of

$$((\frac{1}{N}) * log(W)) = valueof Entropy$$

8: compute cross entropy of $SSID_k$and it should be converges
9: compute kurtosis factor and skew factor of SSID
10: plot entropy, skew factor and kurtosis factor of $SSID_k$ and find threshold values of $entropy_{th}, skew_{th}, andkurtosis_{th}$
11: **If** ((kurtosisFactor ≥ $kurtosis_{th}$) & (kurtosisFactor ≤ $kurtosis_{th}$)) | ((skewFactor ≥ $skew_{th}$) & (skewFactor ≤ $skew_{th}$)) | ((crossEntropy ≥ $entropy_{th}$) & (crossEntropy ≤ $entropy_{th}$))
12: save the dataPoints in $SSID_k$
13: **else**
14: remove data points from $SSID_k$ and save in leftover_dataPoints
15: **EndIf**
16: **EndWhile;**
17: **return**($SSID_k$, $leftover\_SSID_k$)

Figure 23: Subspaces Optimization

Benchmarking Methods The benchmarking or performance of clusters have been evaluated by efficiency assessment. The time complexity with respect to dimensionality of data space, dimensionality of clusters, and scale up and scale down of data object measure the efficiency of assessment.

procedure 3. ADAPTIVE_EXPANSION_REDUCTION_SUBSPACES($SSID_k$, $subspaces_k$, leftover_dataPoints)
Output: $finalSubspaceClusterSSID_k$, $convergence\_SSID_k$, $left\text{-}over\_dataPoints$
If subspaces are of same length
    merge and save simialr $SSID_k$ as $finalSubspaceClusterSSID_k$ and their corresponding subspaces.
EndIf
FOR $k \in \{1, \ldots, finalSubspaceClusterSSID_k\}$
While *not(eachDimension* $\in$ *SSID_k))* **do begin**
    FOR $j \in \{1, \ldots, all\_dimensions\_in\_SSID\_k\}$
    compute convergence of Lindeberg-Feller Central Limit Theorem or Lyapunov Conditions

$$Y_k = (X_k - \mu_k)$$

$$T_k = \Sigma(Y_k)$$

$$S_k^2 = Var(T_k)$$

$$Converge\_SSID_k = T_k \div S_k \to 0$$

    plot $convergence\_SSID_k$ and save the values of convergence of all subspaces corresponding to $SSID_k$
ENDFOR
If ($convergence\_SSID_k = TRUE$)
    save $convergence\_SSID_k$ and optimized subspace "SSID_k as optimized_SSID_k_cluster"
EndIf
EndWhile;
ENDFOR
    return($convergence\_SSID_k$, leftover_dataPoints)

Figure 24: Adaptive Expansion and Reduction of Subspaces

procedure 4. SOFT_CONTEXT_IDENTIFICATION($finalSubspaceClusterSSID_k$)
Output: *softContextDataFrame*
FOR $k \in \{1, \ldots, all\_SSID\}$
    $h_k = soft\_corr\_SSID_k \leftarrow$ compute correlation test of $SSID_k$
    save h_k into "softContextDataFrame"
ENDFOR
    return($softContextDataFrame$)

Figure 25: Soft Context Identification of Subspaces

3 Results

3.1 Effect of SDFS and Existing Standardization Methods on Clustering

We used the nonparametric SAS procedure MODECLUS to cluster the low-dimensional data from experiment one, testing the effects of the different standardization methods. MODECLUS can give clusters of arbitrary shape and size (natural clusters) without constraining their number [31, 49, 50, 74]. The values for the MODECLUS parameters radius (0.20) and threshold (0.25) were tuned until the number and stability of the clusters for the experimental data were optimized. The tuning of MODECLUS parameters results are discussed in the following section. The colors of the points distinguishes the clusters in the three dimensional plots.

3.1.1 Tuning the MODECLUS Parameters θ and r

To understand how many distinct phenotypes might be present in the greenhouse experiment, we used clustering. Since each line has a different basal growth pattern and we wanted to compare among the lines, we first rescaled and standardized the data. We then tuned MODECLUS's parameters threshold, radius, unclassified points and corresponding number of clusters to optimize the number and stability of the clusters. This tuning of the parameters gives the exact number of clusters with zero number of unclassified points, and the clusters predict groups of similar phenotypes. Since parameter tuning is essential for any clustering algorithm, it has become more tolerant for relatively higher speed algorithms. Here, a heuristic search to tune MODECLUS parameters is given and illustrated in Figure 26.

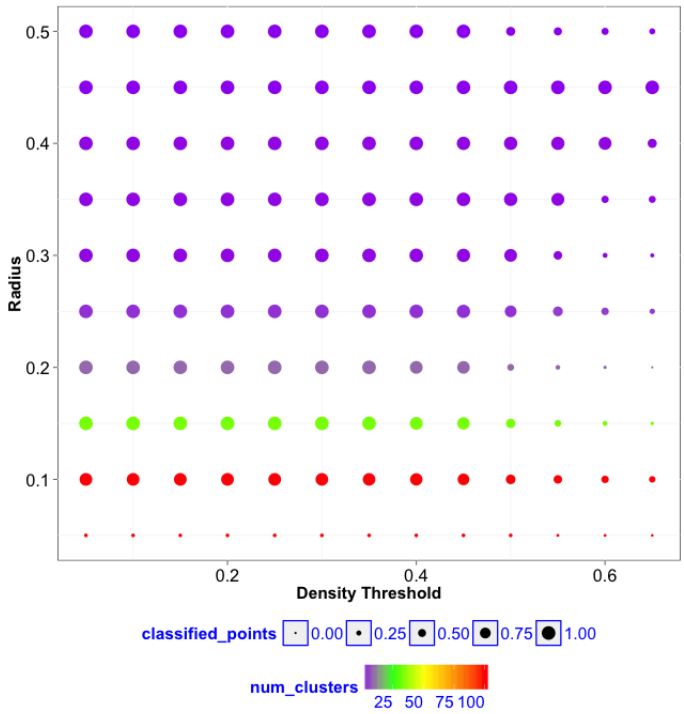


Figure 26: Tuning of Parameters for MODECLUS

Parameter θ Controlling the threshold value of density is the main parameter which decides the lower bounds on the number of clusters that can be obtained. The value of th should be varied in an incremental fashion so as not to increase the number of unclassified points. The value of th can be selected between $0 \leq \theta \leq 1$, and increased by small steps, to avoid unnecessarily large numbers of unclassified points.

Parameter r It controls the radius of sphere such that the limit is decided by θ. The number of clusters obtained and unclassified points changes as value of θ and r change.

3.1.2 Effects of Different Standardization Methods on Clustering

The effects of 18 current standardization techniques are illustrated in Figure 27 and Figure 28.

The effects of different standardization methods on clustering algorithms are summarized in Table 2.

Table 2: Summary of number of clusters using different Standardization Techniques

Methods	**Number of Clusters**	**Number of Unclassified Points**
Mean	5	0
Median	5	0
SDFS	**6**	**0**
Range	10	0
Sum	12	0
Spacing	15	0
Euclen	16	0
ABW	19	0
AHUBER	19	0
AWAVE	19	0
IQR	19	0
MAD	19	0
Maxabs	19	0
USTD	66	0
STD	99	336
AGK	100	341
L	100	579
Mid Range	100	59

These outputs can be grouped into two groups. The first group is like ABW whereas the second group is like SDFS or Mid Range. ABW collapses the clusters into the upper rear corner. Even though SDFS and Mid Range are comparable, their results are very different. Midrange spreads the clusters throughout the volume, while SDFS places the clusters in the center of the space, with the cluster boundaries extending irregularly towards the edges of the volume. Table 2 shows the quality of the clustering by the different methods tested: Mid Range has the highest number of clusters but with relatively few unclassified points, while ABW classifies all points but still produces a large number of clusters. SDFS permits classification of all points, but produces a very moderate number of clusters.

3.1.3 Comparison of Mid Range and SDFS Methods using Synthetic Data

The results of Mid Range and SDFS standardization can be illustrated on synthetic data of Section 2.1.3, and synthetic data were used to verify and compare that SDFS is valid for other datasets. We found SDFS is far better than Mid Range and other standardization methods. Its output is illustrated in Figure 29 and Table 3. The SDFS results are better for these small and sparse datasets, performing better if the number of data objects lies between 3000 and 7000 low dimensional samples. Table 3 shows that the number of unclassified points seems very similar for both methods but the number of clusters are not comparable at all. The Midrange results are not comparable because number of clusters are not more than 100 and these clusters are sparse as well. Midrange seems impractical in terms of groups of similar phenotypes, whereas the relatively low number of clusters found by SDFS are reasonable.

Table 3: Comparison of Number of Clusters Classified by SDFS and MIDRANGE Standardization Techniques

Sample Size	Methods	Number of Clusters	Number of Unclassified Points
3000	Midrange	62	0
3000	SDFS	2	0
6000	Midrange	29	0
6000	SDFS	4	0
7000	Midrange	33	0
7000	SDFS	5	0
10000	Midrange	30	0
10000	SDFS	2	0

3.1.4 Interdependence of Phenotypic Components

Dr. Stapleton's greenhouse experiment evaluated changes in three dimensions of growth in response to stress (Section 2.1.1). To determine how relationships among the data affect clustering with different algorithms and what this can tell us about the number of distinct phenotypes, we compared clustering of raw, standardized, and orthonor-

mally transformed data.

Distributions of the Data Figure 30 shows the lumped distributions for each measured dimension for the raw, standardized, and orthonormally transformed data.

In the raw and orthonormally transformed data, each dimension has a different, non-normal distribution. It is difficult to imagine a single parametric distribution that would fit any dimension of the raw data. Instead, for example, one might mix a normal and a delta distribution. The distributions for Δc and Δs of the orthonormally transformed data appear normal, but the distribution of Δh is very skewed. If one used a distance metric to cluster the orthonormally transformed data using a distance metric, which is very common, the clusters would be at best difficult to interpret because the phenotypic space is warped by the differences in the dimensions.

To determine which phenotypes are shared by different lines and conditions, the data must be standardized in some way to make them comparable, both in relative scales and in distribution. Since each line has a different basal growth pattern and we wanted to compare among the lines, we first linearly rescaled the data and then used SDFS to standardize them [30]; (Vatsa *et al.* , in preparation). The right panel of Figure 30 shows the results: all dimensions have essentially the same nonparametric distribution and the relative scales of the dimensions are the same. Thus, these standardized data are good candidates for clustering.

A common approach to analyzing phenotypic data is to choose one phenotypic dimension of special interest and analyze the relationship of that dimension to some other variate, such as genotype, field location, or weather. In effect, this approach assumes there are no relationships among the phenotypic dimensions: the data for each dimension are functionally univariate and independent of the others. If indeed the dimensions show no or little pairwise covariance, this is a reasonable approach. Otherwise, information can be lost, garbled, or confounded.

To see what can occur if one treats multivariate data as univariate, we synthesized a set of trivariate data where each dimension is linearly independent of the others by transforming the standardized data with an orthonormal basis (Section 2.2.1). If the dimensions are related in some way, this procedure should mangle those relationships when compared to the standardized, untransformed data. To visualize the relationships among the dimensions, we computed the pair-wise covariances for the synthetic, orthonormally transformed data and the standardized data. The results are plotted in Figure 31.

If the data's dimensions are truly mutually independent, then one would see no pattern in their covariances. This is not the case. Compared to the standardized data, the assumption that each dimension can be analyzed independently of the others changes the data and their interpretation in several ways. The planar relationships seen in the standardized data are distorted into a cube (most easily seen in the Δc by Δs covariance). The cube itself is rotated in each pair of dimensions (most easily seen in the Δh by Δc covariance). The data points are compressed, so that their true scatter is lost. The net effect of these changes is to change the pairwise covariances from their values for the standardized data in ways that might be difficult to predict.

Interactions of Clustering Algorithms with Data Transformations We used several algorithms that combine desirable features to cluster the data. We focused on methods that could give clusters of arbitrary shape (*e. g.*, as in many hierarchical clustering algorithms) and that did not constrain the number of clusters (*e. g.*, as in k-means and k-nearest neighbor) [60, 87]. The methods chosen include density-based clustering, using the density of points in regions of the phenotypic space as the similarity criterion, to avoid using distance metrics that apply in only a few types of topological spaces (DBSCAN and OPTCS) [5, 21]; subspace clustering to select the most discriminating combinations of dimensions, to avoid arbitrary dimensional

choices (CLIQUE, SUBCLU, and FIRE) [4, 42, 51]; projected clustering, to avoid the assumption of disjoint, non-overlapping clusters [63]; and nonparametric clustering, to avoid assumptions about the distribution of the data (MODECLUS) [49, 50]. Except for MODECLUS, all of the clustering algorithms we used are found in R packages; there is a SAS procedure for MODECLUS [31, 74].

All clustering algorithms take parameters whose values affect the number of clusters and the number of points falling outside the clusters (the unclassified points"). To select adequate values for the parameters for each algorithm, we explored subsets of the parameter spaces to see how the number of clusters and unclassified points change. For example, density-based clustering algorithms compute the density of the points within spheres of an arbitrary radius. To choose good values for the radius, we computed the number of points that are classified using the k-nearest neighbor algorithm for the standardized and orthonormally transformed data. For each radius, Figure 32 shows the number of classified points. We used radii of 0.17 and 0.022 for the standardized and orthonormally transformed data, respectively.

Similarly, we tuned the MODECLUS parameters (Figure 26) threshold and radius until the number and stability of the clusters was optimized.

How Many Combined Stress Phenotypes Are There? Once the data are comparable and their dimensions have not been arbitrarily omitted, one can now ask how many different phenotypes exist in the population. Not surprisingly, the answer varies as the number of clusters and their membership changes with each clustering algorithm applied. Figure 33 illustrates this reality for the seven clustering algorithms tested, for both the "univariate" data synthesized by the orthonormal transformation and the truly multivariate, standardized data.

The clusters identified by the subspace clustering algorithms can have fewer than three defining dimensions; for these subspaces, the points are plotted along the axes

of those dimensions. One advantage of subspace clustering algorithms is that they classify even points that are missing one or more dimensions, by placing those points in lower dimensional subspaces. This permits the comparison of incomplete data. A disadvantage of CLIQUE is that it doesn't classify points that don't fit the pattern it is discovering.

Figure 33 shows the results of clustering the orthonormally transformed and standardized data using seven different algorithms. The Cartesian spaces for all volumes have been rotated to the same viewpoint for easier comparison. Several trends are immediately obvious.

First, suppressing dimensions distorts the distribution of the data and comprises clustering. DBSCAN, OPTICS, FIRE, and MODECLUS collapse the "univariate" data into a cube. Projected clustering smears the cube of the synthesized data a bit, and SUBCLU finds four clusters, of which two clusters are single points. CLIQUE finds four one-dimensional clusters, two two-dimensional clusters, and one three-dimensional cluster.

Second, neglecting dimensions loses information that can help classify points. For example, DBSCAN finds 8 clusters in the "univariate" data, but cannot classify 1235 points; and 2196 points, nearly all of the remainder, lie in a single cluster. SUBCLU finds four clusters in the "univariate" data, but two of those are clusters of single points. CLIQUE finds seven clusters, of which four contain a reasonable fraction of the points.

Third, most clustering algorithms have difficulty classifying the data into more than one phenotypic group, even when all dimensions of the data are considered. OPTICS, FIRE, and projected clustering lump all the "univariate" points together in a single cluster, whereas DBSCAN, OPTICS, and FIRE find single clusters with the standardized multivariate data. The remaining methods find at least two clusters:

projected clustering, SUBCLUE, and CLIQUE distribute the points fairly unevenly among the clusters, while MODECLUS has a more even distribution of cluster members.

Situations in which an algorithm produces a single cluster tend to have the fewest unclassified points. Thus, DBSCAN finds eight clusters in the orthonormally transformed data and leaves over a third of the points unclassified; but when it finds a single cluster for the standardized points, it places all of the points in that cluster. CLIQUE and SUBCLU find multiple clusters in both datasets, but leave between ≈18–27% of the points unclassified. Both of these subspace clustering procedures are designed for high-dimensional data, which many practitioners assume are parametric. MODECLUS performs best in the sense of finding multiple clusters and leaving no points unclassified: in fact, the algorithm was designed exactly for this situation of nonparametric data.

Table 4 summarizes the number of clusters and unclassified points for each algorithm, for each treatment of the data.

Table 4: Numbers of Clusters and Unclassified Points for Each Algorithm

	number of clusters		*fraction points unclassified*	
algorithm	*orthonormal*	*standardized*	*orthonormal*	*standardized*
DBSCAN	8	1	0.343	0
OPTICS	1	1	5.555×10^{-4}	0
FIRE	1	1	0	0
projected	1	2	0.046	0.343
SUBCLUE	4	4	0.202	0.255
CLIQUE	7	8	0.267	0.188
MODECLUS	1	6	0	0

So how many phenotypes are present in these data? The number of clusters depends on how the data are treated, the relationships among the data, and the choice of clustering algorithm and its parameters. While the true number of clusters and their composition are unknown, the six clusters found by SDFS and MODECLUS

are most consistent with the GWAS analysis of the lines. Those results show eight distinct genotypes that respond differentially to the combined stresses, encompassing six different shapes of response surfaces (Chang *et al.*, in preparation). We conclude SDFS standardization gives results that are more consistent with the likely true number of clusters than the other standardization methods.

The SDFS standardization approach yields nice clusters, but one must determine its applicability to other datasets. In these data, all dimensions are informative in the sense that they change the clusters, but as the dimensionality of data increases, one must pay more attention to the relationships among the dimensions and the informativeness of each dimension and combinations of dimensions.

These kinds of comparative studies can help refine an intuition of the "correct" answer, and both qualitative and quantitative criteria can test the quality of any proposed clusters. For example, MODECLUS returns clusters of arbitrary shape that overlap and interleave with each other, and with appropriate parameter values leave no outlying, unclassified points. Tracking the stability of cluster membership for each datum as a function of parameter and algorithm choice can also persuade one of the quality of a particular classification. As discussed, there are other experimental reasons to believe there are six phenotypes in these data. But in the end, any proposed clustering must be tested against the biology by seeing how well it predicts other related biological phenomena.

3.2 DynaDASC Results

3.2.1 Data Preprocessing

The data preprocessing uses two basic steps in order to assign data points to correct clusters. The first step is normalization. We have been using min-max normalization to make their range equal and this is illustrated by the box-plot in Figure 34(a).

The box-plots in Figure 34(a) illustrate the distributions of the data for each of its nine dimensions (see Section 2.1.4). These dimensions are rescaled between 0 and 1. The histograms of *z_max*, *area_max*, *curvature_max* are unimodal (not shown). *Area_max* and *z_max* have most of zero data points values whereas *curvature_max* has most of the data points near 1. The dimensions *level_0.25*, *level_0.50*, *level_0.75* and $surface\_vol$ have bimodal histograms. These dimensions' data have a mixture of values between zero and one. The frequency of zeroes is greater than the other values in the histogram. The histograms of column and row are multimodal. These dimensions distribution are not normally distributed.

The second step of preprocessing is standardization to make the variances of all dimensions equal such that data points are assigned to correct cluster. We used STD standardization technique and its boxplot is illustrated in Figure 34(b). It shows that the variances of dimensions are similar to each other. The correlation matrix of nine dimensions and the corresponding numeric values are shown in Figure 35 and Table 5 respectively. It is clear that dimensions are not highly correlated to each other. Therefore, we cant apply any dimension reduction technique on this high-dimensional dataset.

	z_max	row	col	surf_vol	L0.75	L0.5	L0.25	curv_max	area_max
z_max	1								
row	0.02	1							
col	-0.12	0.51	1						
surf_vol	-0.07	-0.12	-0.08	1					
L0.75	0.26	0.04	-0.09	-0.01	1				
L0.50	0.26	0.03	-0.11	-0.09	0.5	1			
L0.25	0.21	-0.07	-0.16	-0.14	0.34	0.46	1		
curv_max	-0.31	0.16	0.21	-0.02	-0.34	-0.37	-0.38	1	
area_max	0.09	0.06	0.04	0.01	-0.02	0.02	0.06	-0.16	1

Table 5: Correlation Matrix Numeric Values of Rescaled Dataset

3.2.2 DynaDASC Clustering

The DynaDASC uses five procedures to find subspaces in high-dimensional data (HDD). The first procedure (Figure 22) computes relative adaptive density threshold (τ_k) and assign data objects into subspaces. These subspaces and corresponding data object index values of HDD are returned by this procedure. The second procedure (Figure 23) is subspace optimization function and it takes subspaces as input and optimizes these subspaces based on entropy minimization, skew factor, and kurtosis factor. These three parameters are used to optimize subspaces. The plots of entropy, skew factors, and kurtosis factors are shown in Figure 36, Figure 37, and Figure 38 respectively. This function returns optimized subspaces and the corresponding index values of data objects.

The third procedure (Figure 24) takes input as optimized subspaces and check convergence of each dimensions of every subspace. The convergence has been checked. The Linderberg–Feller Central Limit Theorem or Lyapunov conditions has been applied on all merged and unmerged subspaces and check the convergence of subspaces individually. These subspaces are not converging to zero but very near to zero as shown in Figure 39. This convergence shows that the dimensions are included in subspaces are homogenious to each other and form subspace cluster.

Finally, the optimized and converged clusters index values are used to extract the real data object values. The data object values are used to plot the parallel plot to visualization of clustered subspaces. These subspaces parallel plots are shown in Figure 40.

3.2.3 Benchmarking

The performance of DynaDASC has been measured based on scalability factor versus number of dimensions in subspace. The result illustrated in Figure 41 depicts the

scalability feature of DynaDASC. The DynaDASC discover four subspaces in dataset of size 20000 (pheno1). These subspaces consists of 3, 6, 7, and 9 dimensions in subspace - 4, subspace - 3, subspace - 2, and subspace - 1 respectively. In order to check the scalability feature, we start adding 20000 more data points in existing dataset and we found that the number of dimensions and subspaces change as the data objects increase or decreases in the dataset. The Figure 41 illustrates four dataset results of size 20000 (pheno1), 40000 (pheno2), 60000 (pheno3), and 80000 (pheno4) of data objects. This result conclude that DynaDASC is scalable in nature.

On other hand, DynaDASC is pretty good in finding optimal number of subspace clusters. The existing subspace clustering algorithms discover more than hundred subspaces whereas DynaDASC computes reasonable number of subspace clusters like 4, 4, 4, and 3 for 20000 (pheno1), 40000 (pheno2), 60000 (pheno3), and 80000 (pheno4) of data objects respectively.

The time complexity of DynaDASC has been compared with other existing subspce clustering techniques. The results are depicted in Figure 42. It shows that DynaDASC has $\mathcal{O}(n^2)$ time complexity whereas other have about to linear time complexity except projected clustering algorithm.

4 Discussion, Extensions, and Future Work

4.1 Discussion

Data Standardization Our first experimental study shows that the proposed standardization method, SDFS, performs better than the other techniques evaluated, such as L, Mean, Median, STD, AGK, Euclen, AHUBER, AWAVE, IQR, MAD, Maxabs, USTD, MidRange, Spacing, and Range. This superiority is due to fact that the other methods assume that data have some parametric distribution, while the SDFS technique is very useful for data that has not had any kind of distribution (nonparametric analysis) and sparse datasets with mixtures of positive and negative data values. In contrast to existing standardization techniques, SDFS does not only aim at characterizing low dimensional and sparse data but also will be useful in finding optimization of methods in the low dimensional data analysis process.

A Test for Interdependence of Dimensions If the variates of a multidimensional dataset are genuinely independent, then their clustering should not be affected by transforming the data using an orthonormal basis. This transformation simulates separate, univariate dimensions. This may not be the case for many biological datasets. Indeed, the low-dimensional data of Section 2.1.3 showed bizarre clusters. We found that the clusters are squeezed around the cube and very few data points are allocated to different clusters. Most of the data points are assigned into one cluster. Even after a standardization method such as SDFS is applied on orthogonally transferred data, it still shows the same clustering results. For example, CLIQUE assigns the majority of data points to three subspace clusters and these subspaces are perpendicular to each other, as shown in Figure 33. Therefore, we conclude that the original data are not linearly independent and that an orthonormal transformation generates such artificial linearity. This provides another test, geared towards

clustering, for interdependence of dimensions.

Combining Desirable Features of Other Algorithms A second experiment shows that several diverse clustering algorithms combine desirable features to cluster the data. We focussed on methods that could give clusters of arbitrary shape (for example, as in many hierarchical clustering algorithms) and that did not constrain the numbers of clusters (for example, as in k-means and k-nearest neighbor) [60,87]. The methods chosen include density-based clustering, using the density of points in regions of the phenotypic space as the similarity criterion, to avoid using distance metrics that apply in only a few types of topological spaces (DBSCAN and OPTICS) [5,21]; subspace clustering to select the most discriminating combinations of dimensions, to avoid arbitrary dimensional choices (CLIQUE, SUBCLUE, and FIRES) [4,42,51]; projected clustering, to avoid the assumption of disjoint, non-overlapping clusters [63]; and nonparametric clustering, to avoid assumptions about the distribution of the data (MODECLUS) [49,50]. Except for MODECLUS, all of the clustering algorithms we used are found in R packages; there is a SAS procedure for MODECLUS [31,74].

DynaDASC In this dissertation, we analyzed the major challenges of the density-based subspace clustering problem, and proposed a novel algorithm DynaDASC that does not depend on the characteristics of metric spaces or user inputs for number of clusters. It auto computes and updates the density threshold of each of the subspaces. The density threshold value depends on the measure of locality. It is relatively locally density adaptive, scalable and dynamic in nature. It does not require metrics or any assumptions about the space of the data. This algorithm outperforms existing algorithms in efficiency and effectiveness in detecting the most relevant overlapping and non-disjoint clusters on numerical datasets of lesion phenotypes.

DynaDASC partitions the data space and eventually finds the subspace natural

clusters in any arbitrary subspace even in the presence of overlapping clusters. Using several synthetic and real-world datasets, we demonstrated the advantages of the DynaDASC algorithm compared to the other methods available in the literature for identifying density-based subspace clusters. In contrast to existing clustering techniques, DynaDASC does not only aim at finding clusters but also at identifying the subspaces containing clusters with high accuracy.

DynaDASC was shown to outperform other algorithms for clustering all datasets presented in this study in two important respects. First, the other density-based algorithms placed nearly all of the data points into a single cluster, which we know from other explorations of the data is not correct (data not shown). Second, the subspace clustering algorithms found a very high number of subspaces compared to DynaDASC. Again, other evidence suggests this proliferation of subspaces is probably incorrect. Existing subspace clustering algorithms like Clique, Fires, Subclue, ProClus, Orclus and projected clustering are restricted to cluster spherical or globular shapes, while DynaDASC can discover the natural clusters, whether they are spherical or not. This may explain why DynaDASC finds fewer, but more highly populated clusters. We emphasize that for the large synthetic dataset (Section 2.1.4), we cannot establish the ground truth clustering. Instead, we plan to test the stability of clusters produced by DynaDASC under random partitioning and permutation of data addition (Section 4.3).

However, DynaDASC is clearly not as fast as other subspace clustering algorithms (Figure 42). As expected, DynaDASC runs in quadratic time. Repeatedly recalculating the measures of locality imposes a high computational cost. One obvious solution is to parallelize the computation by randomly partitioning the data, using DynaDASC to cluster each partition, combining identical subspaces across partitions, and recomputing the measures of locality of the entire, preclustered dataset. This approach

could easily exploit distributed cloud resources.

SDFS and DynaDASC are an efficient and effective pair of methods for standardization and detecting the most relevant overlapping subspace clusters on numerical datasets. The experimental study clearly shows that the proposed methods competitively outperform other evaluated techniques. The proposed approach leads to the higher clustering accuracy measured on the basis of the novel standardization method.

4.2 Extensions

Knowledge discovery from big data is a challenging problem in areas such as predictive analytics and wisdom inference. DynaDASC may be very useful and important in many computer vision and image processing problems like high throughput field phenotyping [47], and in the recognition of faces and moving object tracking. Because these datasets are high-dimensional in nature, they need to be efficient with high accuracy clustering techniques. Similarly, face and moving object images of the same object may look entirely different under different illumination conditions and different images can look same under different illumination settings. In online social networks, the detection of communities having similar interests can both aid sociologists and target markets. In social networks, this algorithm could assist in user oriented information integration and provide unique identity, such that data sharing would be managed in a unique way, saving data transmission and cyber security costs as well. In radio astronomy, clusters of galaxies can help cosmologists trace the mass distribution of the universe, and further the understanding of the origin of universe theories.

4.3 Future Work

Several questions remain.

- Does SDFS outperforms methods for multivariate nonparametric standardization, for low- and high-dimensional data?
- DynaDASC uses a parameter τ_k which is the locally density adaptive threshold bounds of the estimating corresponding subspace. Its thresholding of subspace should be an automatic process instead of hard thresholding.
- The optimization process could be enhanced.
- DynaDASC runs in quadratic time: can this be improved without compromising its ability to find natural clusters? For example, by clustering random partitions of the data and then reclustering the combined subspaces and unclassified points.
- The stability of clusters produced by DynaDASC must be investigated by randomly partitioning the data and checking cluster assignment.
- The stability of DynaDASC clusters to random permutation of the order of the input data must be checked.
- How many distinct lesion phenotypes are there? We look forward to applying DynaDASC to our extensive image collection to uncover regions of the causal network.

5 BIBLIOGRAPHY

[1] Achtert, E., Böhm, C., Kriegel, H., Kröger, P., Müller, I., and A. Zimek, 2007. Detection and visualization of subspace cluster hierarchies. In *12th International Conference on Database Systems for Advanced Applications*, pages 152–163. Springer Verlag, Bangkok Thailand.

[2] Agarwal, R., Gehrke, J., Gunopulos, D., and P. Raghavan, 2005. Automatic Subspace Clustering of High Dimensional Data. *Data Mining and Knowledge Discovery* **11**:5–33.

[3] Aggarwal, C. C. and C. K. Reddy, 2014. *DATA CLUSTERING Algorithms and Applications*. CRC Press, Inc., New York.

[4] Agrawal, R., Gehrke, J., Gunopulos, D., and P. Raghavan, 2005. Automatic Subspace Clustering of High-Dimensional Data. *Data Mining and Knowledge Discovery* **11**:5–33.

[5] Ankerst, M., Breunig, M. M., and J. S. Hans-Peter Kriegel, 1999. OPTICS: ordering points to identify the clustering structure. In Delis et al. [19], pages 49–60.

[6] Araus, J. L. and J. E. Cairns, 2014. Field high-throughput phenotyping: the new crop breeding frontier. *Trends Plant Sci.* **19**:52–61.

[7] Art, D., Gnanadesikan, R., and J. Kettenring, 1982. *Data-Based Metrics for Cluster Analysis*. Utilitas Mathematica, New York.

[8] Aziz, M. S. and C. K. Reddy, 2010. A robust seedless algorithm for correlation clustering. In Zaki, M. J., Yu, J. X., Ravindran, B., and V. Pudi, eds., *Advances*

in Knowledge Discovery and Data Mining, number 14 in Advances in Knowledge Discovery and Data Mining, pages 28–37. Springer Verlag, Berlin.

[9] Bellman, R. E., 1961. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, Princeton NJ.

[10] Beyer, K., Goldstein, J., Ramakrishnan, R., and U. Shaft, 1998. An entropy weighting mixture model for subspace clustering of high–dimensional data. In Beeri, C. and P. Buneman, eds., *Proceedings, Lecture Notes in Computer Science 1540*, pages 217–235. Springer Verlag, Berlin.

[11] Bickel, P. J., 1975. One-Step Huber Estimates in the Linear Model. *J. Am. Stat. Assoc.* **70**:428–434.

[12] Bluma, A. L. and P. Langley, 1997. Selection of relevant features and examples in machine learning. *artificial intelligence* **97**:245–271.

[13] Böhm, C., Kailing, K., Kriegel, H., and P. Kröger, 2004. Density connected clustering with local subspace preferences. In *ICDM 4th IEEE Int. Conf. on Data Mining, November 01–04, 2004, Brighton, UK*, pages –. IEEE, Brighton, UK.

[14] Borchers, H. W., 2015. R package `pracma`: Practical numerical math functions. Technical report, Comprehensive R Archive Network.

[15] Capaldi, A. P., Kaplan, T., Liu, Y., Habib, N., Regev, A., Friedman, N., and E. K. O'Shea, 2008. Structure and function of a transcriptional network activated by the MAPK Hog1. *Nature Genet.* **40**:1300–1306.

[16] Chu, Y., Huang, J., Chuang, K., and D. Yang, 2010. Density Conscious Subspace Clustering for High-Dimensional Data. *IEEE Trans. Know. Data Eng.* **22**:16–30.

[17] Coffin, D., 2016–present. *Decoding Raw Digital Photos in Linux*. David Coffin, `https://www.cybercom.net/ dcoffin/dcraw/`.

[18] D. C. Hoaglin, F. M. and J. W. Tukey, eds., 1983. *Understanding Robust and Exploratory Data Analysis*, New York. John Wiley and Sons.

[19] Delis, A., Faloutsos, C., and S. Ghandeharizadeh, eds., 1999. *Proceedings ACM SIGMOD International Conference on Management of Data*, New York. Association for Computing Machinery Press.

[20] Domeniconi, C., Gunopulos, D., Ma, S., Papadopoulos, D., and B. Yan, 2007. Locally Adaptive Metrics for Clustering High Dimen- sional Data. *Data Mining and Knowledge Discovery* **14**:63–97.

[21] Ester, M., Kriegel, H., Sander, J., and X. Xu, 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In Simoudis, V., Han, J., and U. Fayyad, eds., *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining*, pages 1–6. American Association for Artificial Intelligence.

[22] Fauvel, M., Chanussot, J., Benediktsson, J. A., and A. Villa, 2013. Parsimonious Mahalanobis kernel for the classification of high dimensional data. *Pattern Recognition* **46**:845–854.

[23] Friedman, J. H., 1994. Estimating functions of mixed ordinal and categorical variables using adaptive splines. In Morgenthaler, S., Ronchetti, E., and W. Stahel, eds., *New Directions in Statistical Data Analysis and Robustness*, Monte Verità Proceedings of the Centro Stefano Franscini, Ascona. Birkhäuser, New York.

[24] Friedman, N., 2004. Inferring cellular networks using probabilistic graphical models. *Science* **303**:799–805.

[25] Friedman, N., Linial, M., Nachman, I., and D. Pe'er, 2000. Using Bayesian networks to analyze expression data. In *RECOMB2000. Proceedings of the 2000 RECOMB Meeting, Tokyo*, pages 127–135. Association for Computing Machinery, New York.

[26] Friedman, N., Linial, M., Nachman, I., and D. Pe'er, 2000. Using Bayesian networks to analyze expression data. *J. Comp. Biol.* **7**:601–620.

[27] Gao, B. J., Griffith, O. L., Ester, M., and S. J. M. Jones, 2006. Discovering significant opsm subspace clusters in massive gene expression data. In Ungar, L., Craven, M., Gunopulos, D., and T. Eliassi-Rad, eds., *Proceedings of the 12th acm sigkdd international conference on knowledge discovery and data mining*, pages 922–928. Association for Computing Machinery, Philadelphia.

[28] Goodall, C., 1983. M-estimators of location: An outline of theory. In D. C. Hoaglin and Tukey [18], pages 339–403.

[29] Guan, Y., Myers, C. L., Lu, R., Lemischka, I. R., Bult, C. J., and O. G. Troyanskaya, 2008. A genomewide function network for the laboratory mouse. *PLoS Comput. Biol.* **4**:e1000165.

[30] Han, J. and M. Kamber, 2006. *Cluster Analysis*. Morgan Kaufmann, New York, second edition.

[31] Hassani, M., Hansen, M., Müller, E., Assent, I., Günnemann, S., Jansen, T., and T. Seidl, 2015. R package subspace: Interface to opensubspace. Technical report, Comprehensive R Archive Network.

[32] Hinneburg, A. and H. Gabriel, 1998. DENCLUE 2.0: fast clustering based on kernel density estimation. In *The Fourth International Conference on Knowledge Discovery and Data Mining*, pages 1–11. American Association for Artificial Intelligence, New York.

[33] Hinneburg, A. and D. A. Keim, 1999. Optimal grid clustering towards breaking the curse of dimensionality in high dimensional clustering. In Atkinson, M. P., Orlowska, M. E., Valduriez, P., Zdonik, S. B., and M. L. Brodie, eds., *Proceedings of 25th International Conference on Very Large Data Bases*, pages 1–12. Morgan Kaufmann.

[34] hung Cheng, C., thee Fu, A. W., and Y. Zhang, 1999. Entropy–based subspace clustering for mining numerical data. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 84–93. Association for Computing Machinery Press, California.

[35] Ideker, T., Thorsson, V., Ranish, J. A., Christmas, R., Buhler, J., Eng, J. K., Bumgarner, R., Goodlett, D. R., Aebersold, R., and L. Hood, 2001. Integrated genomic and proteomic analyses of a systematically perturbed metabolic network. *Science* **292**:929–934.

[36] Iglewicz, B., 1983. Robust scale estimators and confidence intervals for location. In D. C. Hoaglin and Tukey [18], pages 405–431.

[37] Jajuga, K. and M. Walesiak, 2000. Standardization of Data Set Under Different Measurement Scales. *Chapter Classification and Information Processing at the Turn of the Millennium Part of the series Studies in Classification, Data Analysis, and Knowledge Organization* **1**:105–112.

[38] Jing, L., Ng, M. K., and J. Z. Huang, 2007. An Entropy Weighting k-Means Algorithm for Subspace Clustering of High-Dimensional Sparse Data. *IEEE Trans. Know. Data Eng.* **19**:1026–1041.

[39] Johal, G. S., 2007. Disease lesion mimic mutants of maize. *APSnet* **July**:`http://-www.-apsnet.-org/-online/-feature/-mimics/-default.--asp`.

[40] Jolliffe, I. T., 2002. *Principal Component Analysis*. Springer Verlag, New York, 2nd edition.

[41] Kafadar, K., 1982. The Efficiency of the Biweight as a Robust Estimator of Location. *J. Research of the National Bureau of Standards* **88**:105–116.

[42] Kailing, K., Kriegel, H., and P. Kröger, 2004. Density-connected subspace clustering for high-dimensional data. In *4th International Conference on Data Mining*, pages 246–257. Society for Industrial and Applied Mathemetics, Philadelphia.

[43] Kaufman, L. and P. J. Rousseeuw, 2005. *Finding Groups in Data*. John Wiley and Sons, Inc., New York, first edition.

[44] Kaur, A. and A. Datta, 2015. A novel algorithm for fast and scalable subspace clustering of high–dimensional data. *J. Big Data.* **2**:1–24.

[45] Kazic, T., 2015. Ten simple rules for experiments' provenance. *PLoS Comput. Biol.* **11**:e1004384.

[46] Kelly, D., Vatsa, A., Mayham, W., and T. Kazic, 2015. Extracting complex phenotypes from images. *Mach. Vision Appl.* page (in press).

[47] Kelly, D., Vatsa, A., Mayham, W., Ngô, L., Thompson, A., and T. Kazic, 2015. An Opinion on Imaging Challenges in Phenotyping Field Crops. *Mach. Vision Appl.* page (in press).

[48] Koller, D. and N. Friedman, 2009. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, Cambridge MA.

[49] Koontz, W. L. and K. Fukunaga, 1972. Asymptotic analysis of a nonparametric clustering technique. *IEEE Computer* **21**:967–974.

[50] Koontz, W. L. G. and K. Fukunaga, 1972. A nonparametric valley-seeking technique ror cluster onalysis. *IEEE Computer* **21**:171–178.

[51] Kriegel, H., Kröger, P., Renz, M., and S. Wurst, 2005. A generic framework for efficient subspace clustering of high-dimensional data. In *ICDM 5th IEEE International Conference on Data Mining, November 27–30, 2005, Houston, Texas, USA*, pages 1–8. IEEE, Houston, TX.

[52] Kriegel, H.-P., Kröger, P., and Arthurzimek, 2009. Clustering high-dimensional data: a survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Trans. Know. Disc.* **3**:1–58.

[53] Kriegel, H.-P., Króger, P., and A. Zimek, 2008. Detecting Clusters in Moderate-to-High Dimensional Data: subspace clustering, pattern-based clustering, and correlation clustering. In Jagadish, H. V., ed., *Proceedings of the VLDB Endowment*, volume 1 of *2008*, pages 1–2. Association for Computing Machinery Press, Auckland New Zealand.

[54] Liu, H. and H. Motoda, 1998. *Feature Selection for Knowledge Discovery and Data Mining*. Springer Verlag, New York.

[55] Liu, L. and L. Wang, 2013. A scalable unsupervised feature merging approach to efficient dimensionality reduction of high dimensional visual data. In *IEEE International Conference on Computer Vision*, pages 3008–3015. IEEE, Sydney.

[56] Lloyd, S. P., 1982. Least squares quantization in pcm. *IEEE Trans. Info. Theory* **28**:129–137.

[57] Lynch, J. P., 2011. Root phenes for enhanced soil exploration and phosphorus acquisition: tools for future crops. *Plant Physiol.* **156**:1041–1049.

[58] Lynch, J. P., 2015. Root phenes that reduce the metabolic costs of soil exploration: opportunities for 21st century agriculture. *Plant Cell Environ.* **38**:1775–1784.

[59] MacQueen, J. B., 1967. Some methods for classification and analysis of multivariate observations. In Cam, L. M. L. and J. Neyman, eds., *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297. University of California Press.

[60] Marrelec, G., Messé, A., and P. Bellec, 2015. A Bayesian alternative to mutual information for the hierarchical clustering of dependent random variables. *PLoS One* **10**:1–26.

[61] Mohebi, A., Aghabozorgi, S., Wah, T. Y., Herawan, T., and R. Yahyapour, 2016. Iterative big data clustering algorithms: a review. *SOFTWARE: PRACTICE AND EXPERIENCE* **46**:107–129.

[62] Moise, G., Sander, J., and M. Ester, 2006. P3C: a robust projected clustering algorithm. In *Sixth International Conference on Data Mining*, pages 1–12. IEEE, Hong Kong, China.

[63] Müller, E., Günnemann, S., Assent, I., and T. Seidl, 2009. Evaluating clustering in subspace projections of high dimensional data. In *Very Large Data Base Endowment*, pages 1–12. Association for Computing Machinery, Lyon, France.

[64] Nagesh, H., Goil, S., and A. Choudhary, 2001. Adaptive grids for clustering massive data sets. In Kumar, V. and R. Grossman, eds., *Proceedings of the 2001 SIAM International Conference on Data Mining*, pages 1–17. Society for Industrial and Applied Mathematics, Auckland New Zealand.

[65] Neuffer, M. G., Edward H. Coe, Jr., and S. R. Wessler, 1997. *Mutants of Maize*. Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY.

[66] Neuffer, M. G., Hoisington, D., and V. Walbot, 1985. The lesion mutants of maize. In Freeling, M., ed., *Plant Genetics*. Alan R. Liss, New York.

[67] Owen, M., 2010. *Tukey's Biweight Correlation and the Breakdown*. Master's thesis, Pomona College, California.

[68] Parsons, L., Haque, E., and H. Liu, 2004. Subspace Clustering for High Dimensional Data: A Review. *SIGKDD Explorations* **6**:90–105.

[69] Pe'er, D., Regev, A., Elidan, G., and N. Friedman, 2001. Inferring subnetworks from perturbed expression profiles. *Bioinformatics* **17**:S215–S224.

[70] Peña, J. M., Lozano, J. A., Larrañaga, P., and I. Inza, 2001. Dimensionality reduction in unsupervised learning of conditional gaussian networks. *IEEE Trans. Patt. Anal. Mach. Intell.* **23**:590–603.

[71] Peng, L. and J. Zhang, 2011. An entropy weighting mixture model for subspace clustering of high-dimensional data. *Pattern Recognition* **32**:1154–1161.

[72] Procopiuc, C. M., Jones, M., Agarwal, P. K., and T. M. Murali, 2002. A monte carlo algorithm for fast projective clustering. In *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, pages 418–427. Association for Computing Machinery Press, New York.

[73] REHIOUI, H., IDRISSI, A., ABOUREZQ, M., and F. ZEGRARI, 2016. DENCLUE-IM: A New Approach for Big Data Clustering. *Procedia Computer Science* **83**:560–567.

[74] SAS 9.3, 2015–present. *MODECLUS Procedure*. SAS/STAT(R) 9.3 User's Guide, `http://support.sas.com/documentation/cdl/en/statug/63033/HTML/default/viewer.htm#modeclus_toc.htm`.

[75] Sembiring, R. W. and J. M. Zain, 2010. Cluster evaluation of density based subspace clustering. *J. Computing* **2**:1–6.

[76] Sheikholeslami, G., Chatterjee, S., and A. Zhang, 1998. Wavecluster: a multi resolution clustering approach for very large spatial databases. In *Proceedings of the International Conference on Very Large Databases (Vldb)*, pages 1–12. Morgan Kaufmann.

[77] Shirkhorshidi, A. S., Aghabozorgi, S., Wah, T. Y., and T. Herawan, 2014. Big data clustering: A review. In *Computational Science and Its Applications - ICCSA 2014*, pages 707–720. Int. Conf. on Computational Science and Its Applications, Guimaraes, Portugal.

[78] Singh, V. and S. Laxman, 2013. Subspace clustering of high-dimensional data: an evolutionary approach. *Comp. Intel. and Soft. Comp. J.* **2013**:1–12.

[79] Steinbach, M., Tan, P.-N., and V. Kumar, 2004. Support envelopes: a technique for exploring the structure of association patterns. In Piatetsky, G., ed.,

Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 1–10. Association for Computing Machinery, New York.

[80] Sullivan, J. M., 2006. Curvature measures for discrete surfaces. In Grinspun, E., Desbrun, M., Polthier, K., and P. Schröder, eds., *Discrete Differential Geometry. An Applied Introduction*, pages 10–13. Columbia University, `http://ddg.cs.columbia.edu/SIGGRAPH06/DDGCourse2006.pdf`.

[81] Wang, W., Yang, J., and R. R. Muntz, 1997. STING: a statistical information grid approach to spatial data mining. In *Proceedings of the 23rd international conference on very large data bases*, pages 186–195. Association for Computing Machinery, California.

[82] Woo, K.-G., Lee, J.-H., Kim, M.-H., and Y.-J. Lee, 2004. FINDIT: a fast and intelligent subspace clustering algorithm using dimension voting. *Information and Software Technology* **46**:255–271.

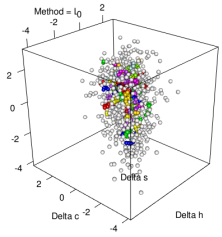
[83] Xia, H., Zhuang, J., and D. Yu, 2013. Novel soft subspace clustering with multi-objective evolutionary approach for high-dimensional data. *Pattern Recognition* **46**:2562–2575.

[84] Xu, M., Chen, H., and P. k. Varshney, 2013. Dimensionality reduction for registration of high-dimensional data sets. *IEEE Trans. Image Proc.* **22**:3041–3049.

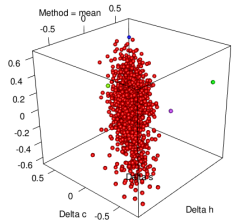
[85] Yiu, M. L. and N. Mamoulis, 2003. Frequent-pattern based iterative projected clustering. In Wu, X., Tuzhilin, A., and J. W. Shavlik, eds., *Proceedings of the Third IEEE International Conference on Data Mining, 19–22 November, 2003, Melbourne*, pages 1–4. IEEE, Los Alamitos, CA.

[86] Yu, L. and H. Liu, 2003. Feature selection for high-dimensional data: a fast correlation-based filter solution. In Fawcett, T. and N. Mishra, eds., *Proceedings of the twentieth international conference on machine learning*, pages 1–8. American Association for Artificial Intelligence, California.

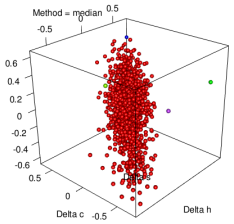
[87] Zhu, X. and D. R. Hunter, 2015. Clustering via finite nonparametric ICA mixture models. *arXiv:1510.08178v2* pages 1–27.



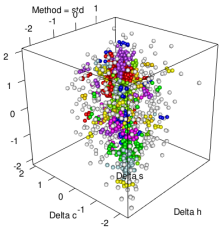
L



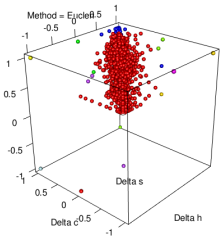
Mean



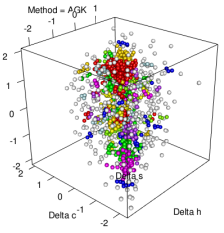
Median



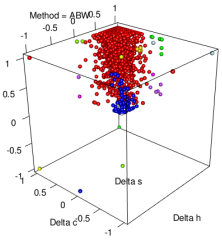
STD



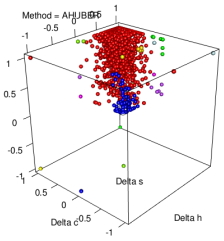
Euclen



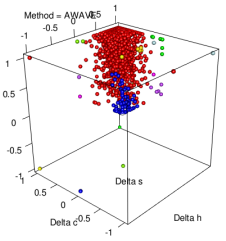
AGK



ABW

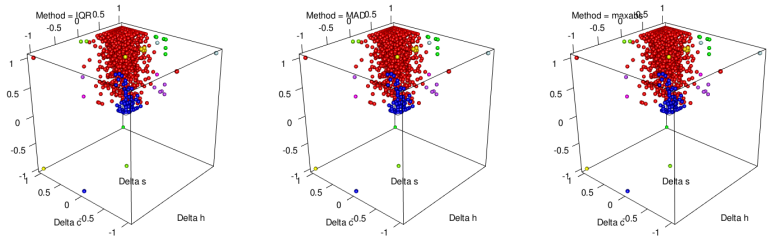


Ahuber

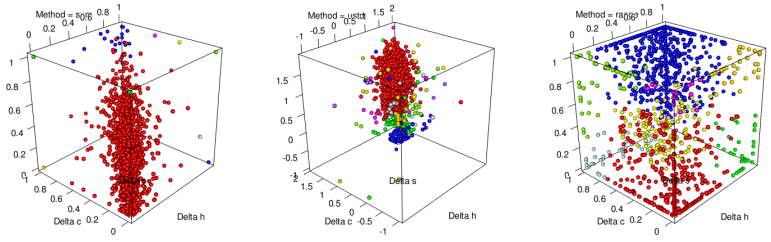


Awave

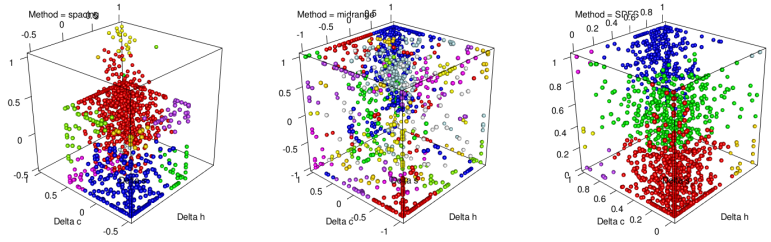
Figure 27: Clustering Output of Standardized Data



IQR MAD Maxabs



Sum USTD Range



Spacing Midrange SDFS

Figure 28: Clustering Output of Standardized Data

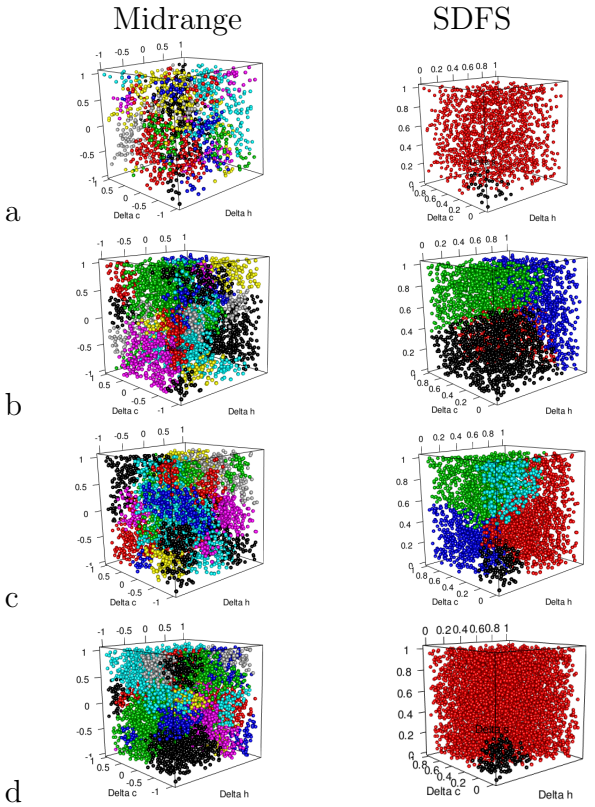


Figure 29: Modeclus Results of Synthetic Data using MIDRANGE and SDFS (a) Size : 3000, (b) Size : 6000, (c) Size : 7000, and (d) Size : 10000

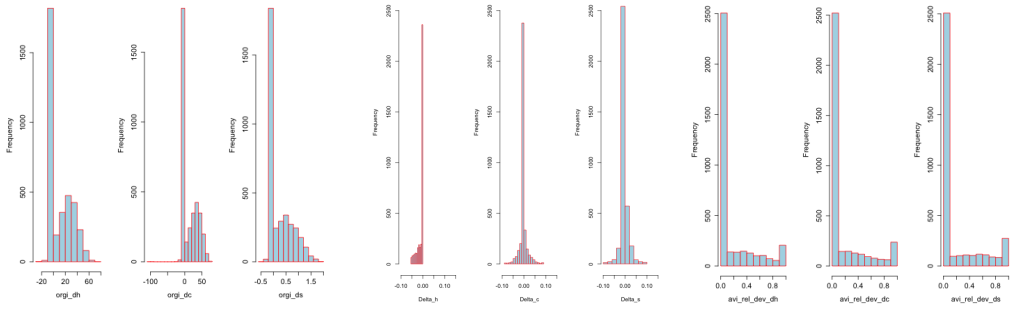


Figure 30: Distributions of Raw, Orthonormally Transformed, and Standardized Data

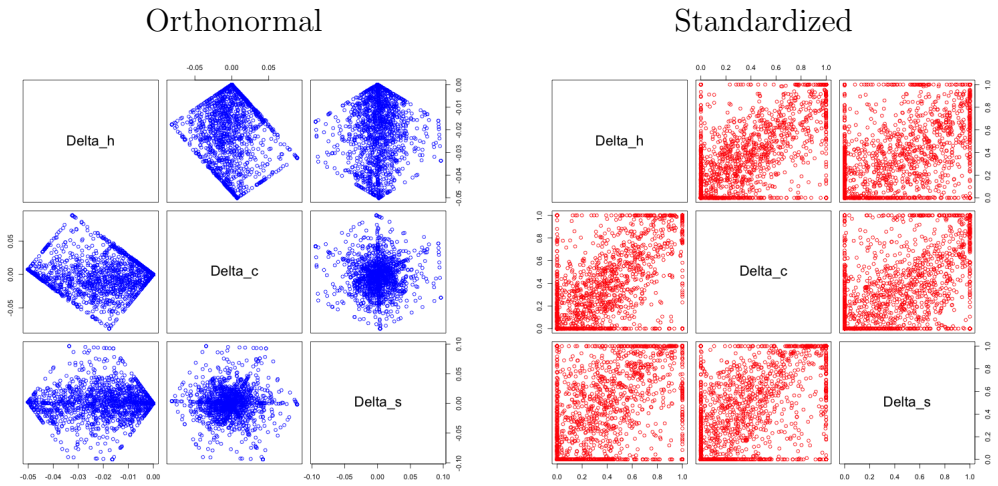


Figure 31: Pair-Wise Covariances of Orthonormally Transformed and Standardized Data

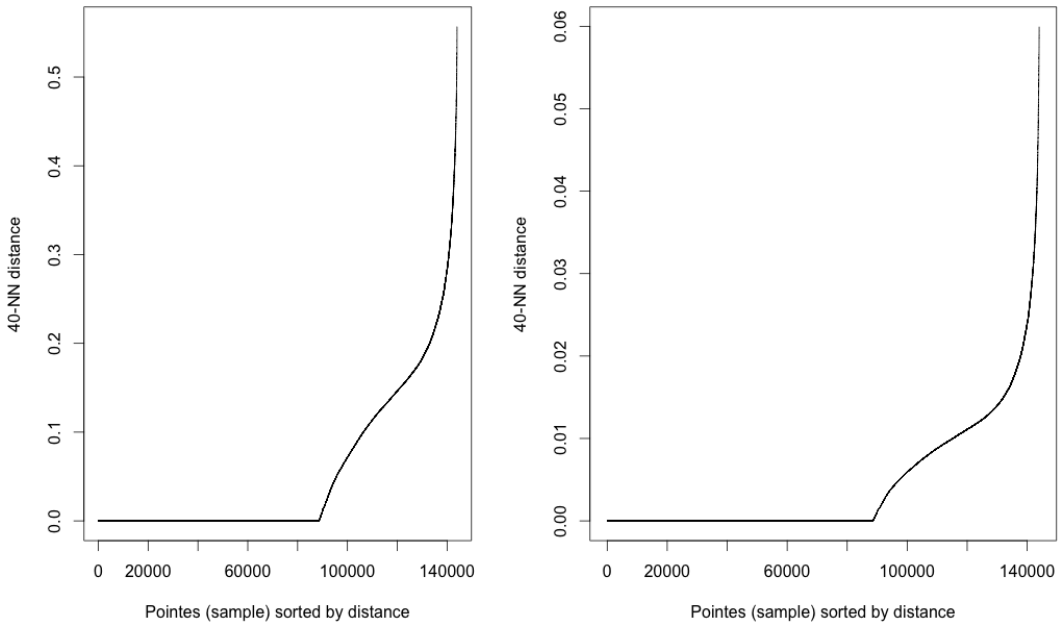


Figure 32: Points/cluster for Spheres of Increasing Radius. Left, standardized data; right, orthonormally transformed data.

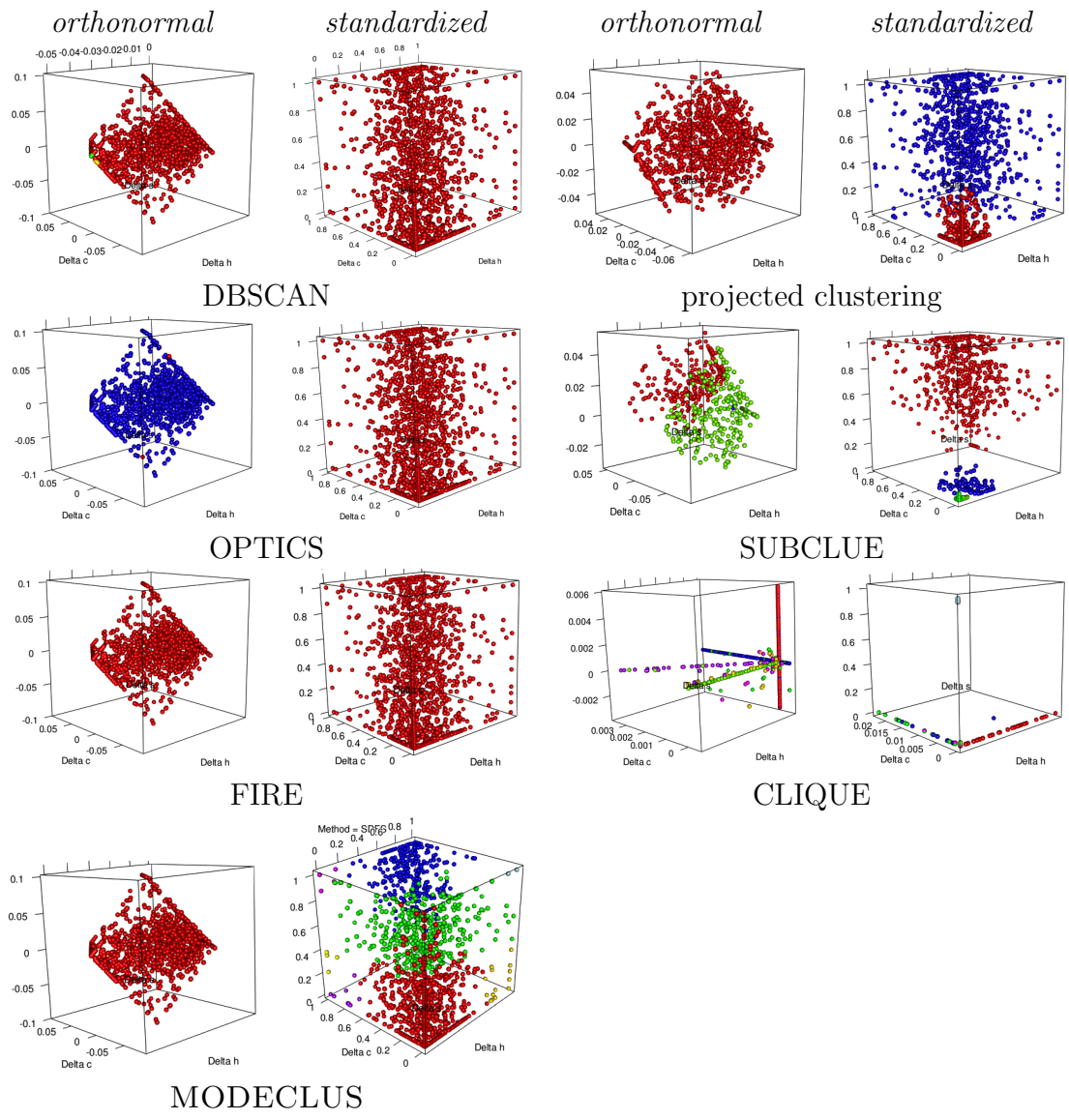
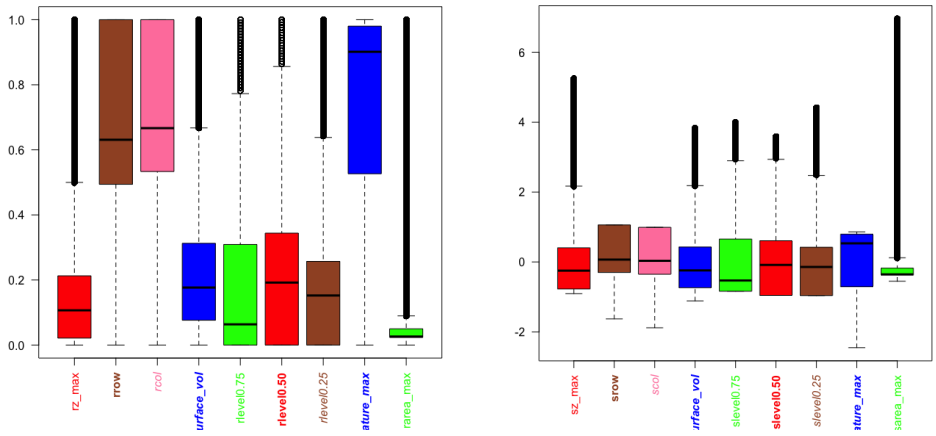


Figure 33: Clusters Using Different Algorithms on Orthonormally Transformed and Standardized Data



(a) Rescaled (b) Standardized

Figure 34: Box Plots of Rescaled and Standardized High-Dimensional Synthetic Data

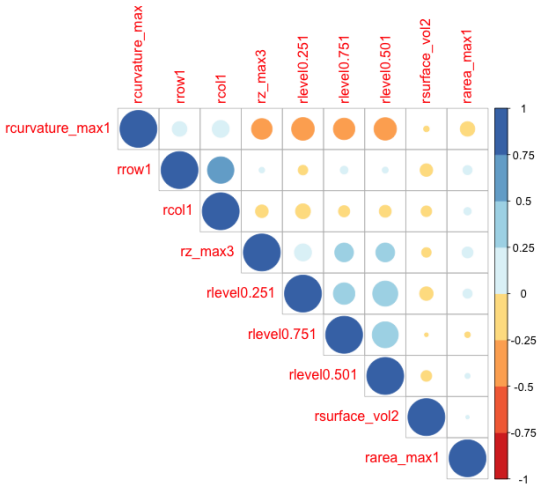
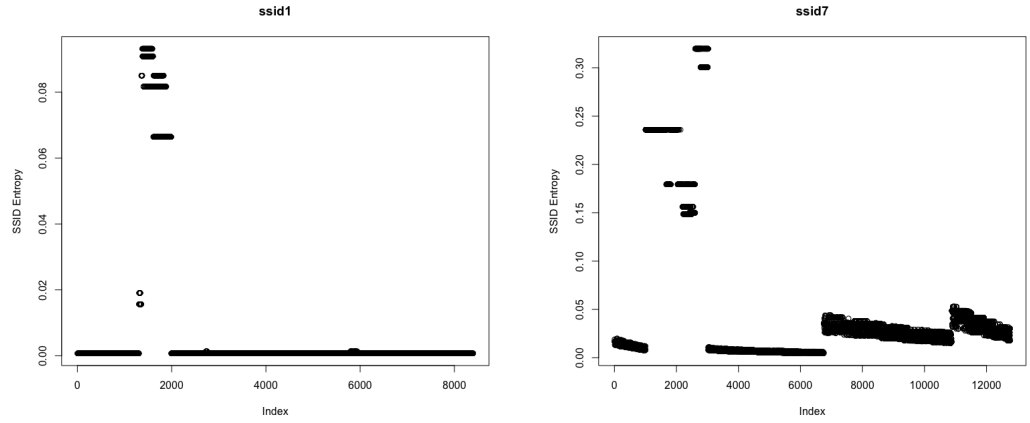
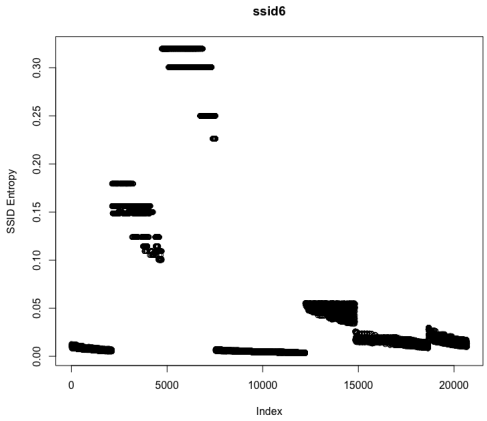


Figure 35: Correlation Matrix Plot

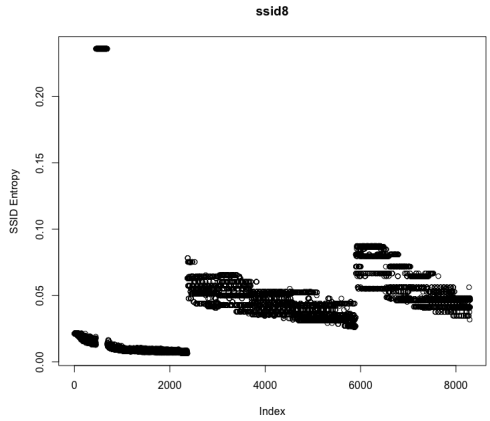


subspace1

subspace2

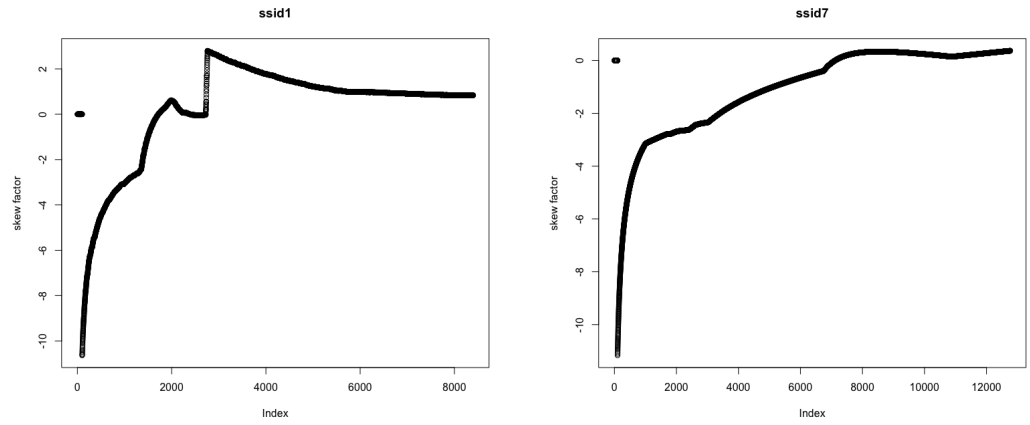


subspace3

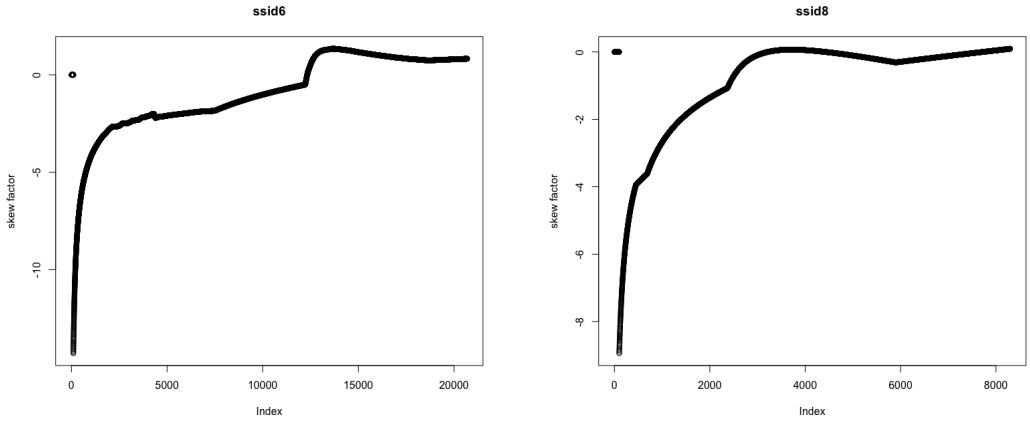


subspace4

Figure 36: Subspace Optimization: Entropy Thresholding

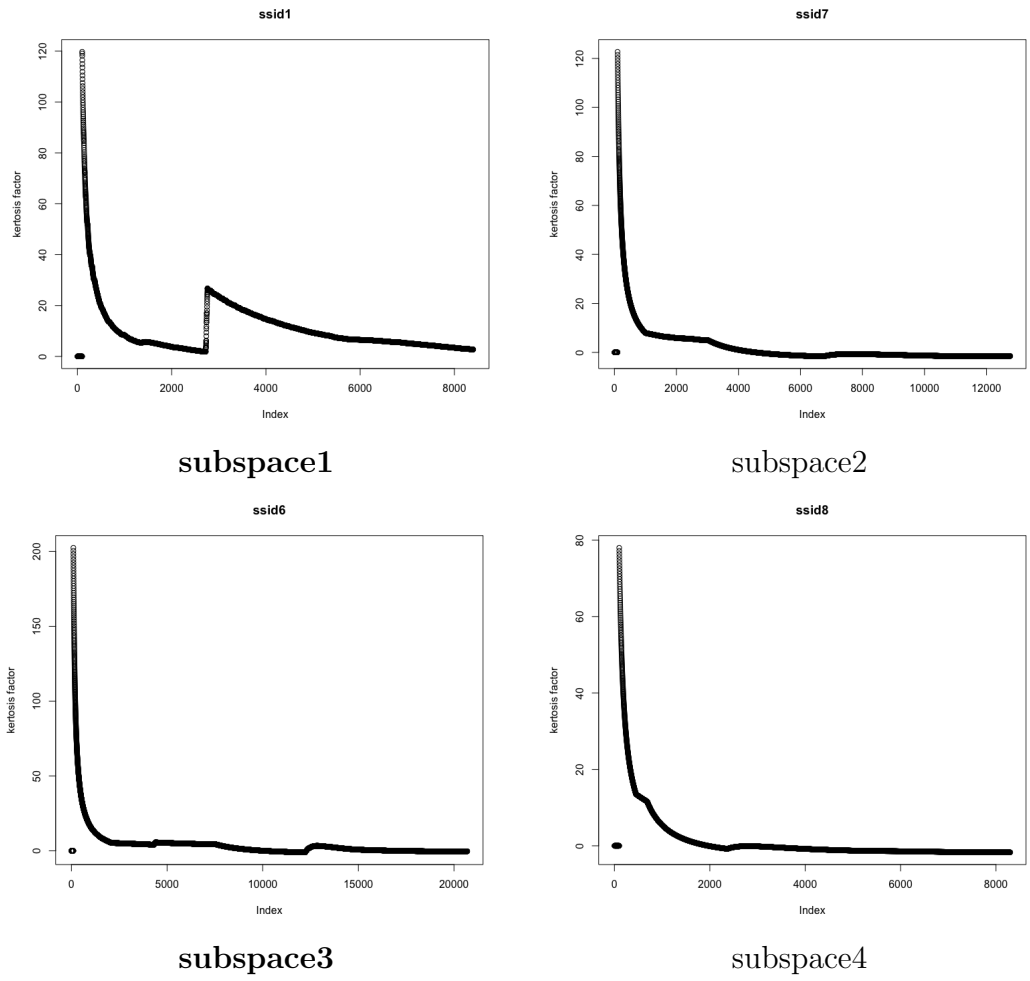


subspace1 subspace2



subspace3 subspace4

Figure 37: Subspace Optimization: Skew Factor Thresholding



subspace1 subspace2

subspace3 subspace4

Figure 38: Subspace Optimization: Kurtosis Factor Thresholding

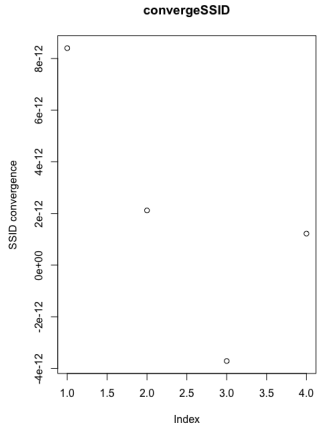


Figure 39: Subspaces' Convergence Points

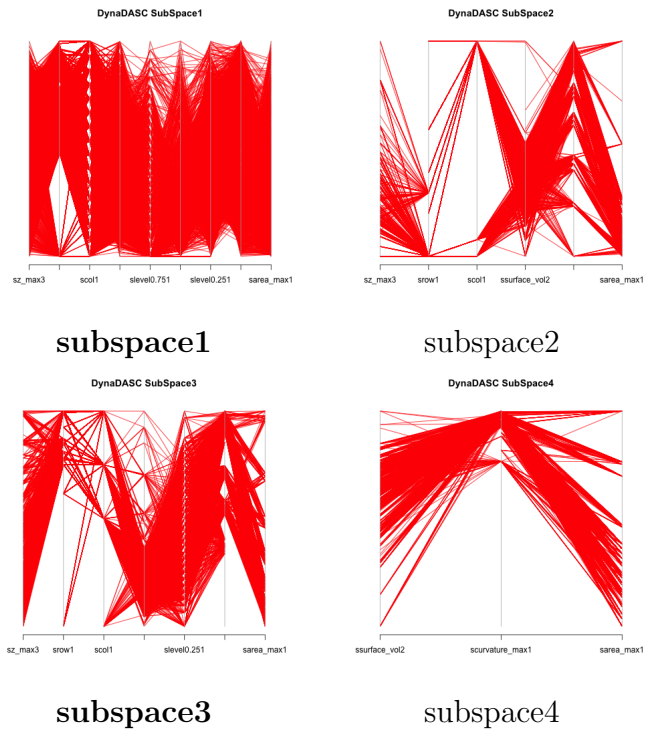


Figure 40: Parallel Plot of Subspace Cluster

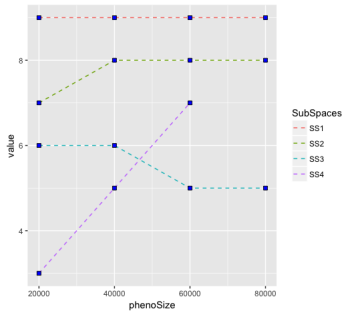


Figure 41: Number of Dimensions in Subspaces as a Function of Scalability

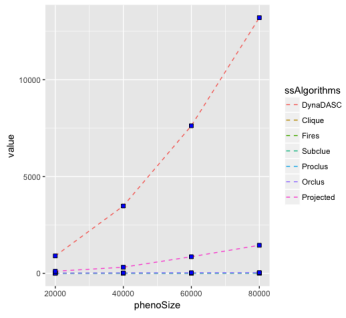


Figure 42: Comparision of Time Complexity of DynaDASC to Other Algorithms

VITA

Avimanyou Kumar Vatsa is going to join as a tenure track Assistant Professor at West Texas A&M University from fall 2017. He is a Ph.D. candidate and received MS (Computer Science) and minor in statistics from University of Missouri - Columbia, MO, USA in 2015. He obtained his M.Tech. (Computer Engineering) with Hons. from Shobhit University, Meerut and B.Tech. (Information Technology) from V.B.S. Purvanchal University, Jaunpur, (U.P), INDIA, in 2009 and 2001 respectively. He is working as a teaching and research assistant at University of Missouri Columbia, MO USA and worked as an Assistant Professor for more than ten years in several engineering colleges and university in INDIA. He had been member of academic and administrative bodies. During his teaching, he had been coordinated many technical fests and national conference at college and university level. He has worked as software engineer in software industry. He is on the editorial board member and reviewers of several international and national journals in computational biology, bioinformatics, networks and security field. His area of research includes Low and High Dimensional Data Analytics, Image Processing, MANET (Mobile Ad-Hoc Network), Computer Network and Cyber Security.