2013

Scalable Techniques for the Analysis of Large-scale Materials Data

Sai Kiranmayee Samudrala
Iowa State University

Scalable techniques for the analysis of large-scale materials data

by

Samudrala Sai Kiranmayee

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Mechanical Engineering

Program of Study Committee:

Baskar Ganapathysubramanian, Co-Major Professor

Krishna Rajan, Co-Major Professor

Srinivas Aluru

Sriram Sundararajan

Ross Morrow

Jaroslaw Zola

Iowa State University

Ames, Iowa

2013

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

This thesis would not have been written without the valuable assistance of the people who contributed towards my research in one way or the other. I am greatly indebted and would like to thank them for all their support.

First and foremost, I would like to sincerely thank my advisor Dr. Baskar Ganapathysubramanian for his guidance and constant encouragement. He has been a true inspiration and a great support as I leaped across the various hurdles in every phase of my research.

I would also like to thank Dr. Krishna Rajan and Dr. Srinivas Aluru for their timely guidance which helped me in my research. My sincere thanks to my Program of Study committee members, Dr. Sriram Sundararajan and Dr. William Ross Morrow for their guidance and comments. Dr. Jaroslaw Zola and Dr. Olga Wodo were extremely helpful with their insightful comments and stimulating discussions which helped me in striving for perfection in my research.

I would like to thank my fellow colleagues for their constant support, their sincere interest in my work and inclination to helping me which led to creative techniques for resolving issues in my research.

Last but not the least, my family for their love, patience, and the belief in my ability to reach my goals. They have been my strength as I journeyed through my doctorate education far away from home in this distant land.

DEDICATION

I would like to dedicate this thesis to my family. I would like to thank my father, Gopal Rao Samudrala and mother, Anita Rani Kanala, for their constant support throughout my life. I would also like to thank my sister Sai Karuna Samudrala and brother-in-law, Sudhir Kovur for being with me throughout the journey of my student life. This acknowledgement will be incomplete without my gratitude towards my husband, Aditya Kunduri and my mother-in-law, Padma Kunduri for their sincere faith and encouragement.

ABSTRACT

Many physical systems of fundamental and industrial importance are significantly affected by the development of new materials. By establishing process-structure-property relationship one can design new, tailor-made materials that possess desired properties. Conventional experimental and analytical techniques like first-principle calculations, though accurate, are extremely tedious and resource-intensive resulting in a significant gap between the time of discovery of a new material and the time it is put to engineering practice. Furthermore, huge amounts of data produced by these techniques poses a tough challenge in terms of analysis. This thesis addresses the challenges in analyzing huge datasets by leveraging the advanced mathematical and computational techniques in order to establish process-structure-property relationship of materials.

First of the three parts of this thesis describes application of dimensionality reduction (DR) techniques to analyze a dataset of apatites ($A_4^I A_6^{II}(BO_4)_6X_2$) described in structural descriptor space. This data reveals interesting correlations between structural descriptors like ionic radius and covalence with characteristic properties like apatite stability; information crucial to promote the use of apatites as an antidote in lead poisoning. Second part of the thesis describes a parallel spectral DR framework that can process thousands of points lying in a million dimensional space, which is beyond the reach of currently available tools. To further demonstrate applicability of our framework we perform dimensionality reduction of 75,000 images representing morphology evolution during manufacturing of organic solar cells in order to identify the optimal processing parameters. Third significant approach discussed in this thesis includes applying well-studied graph-theoretic methods to analyze large datasets produced from Atom Probe Tomography (APT) to quantify the morphology of precipitates in a solvent material. The above three mathematical models and computational strategies were applied to large-scale materials data in order to establish process-structure-property relationship.

CHAPTER 1. INTRODUCTION

1.1 Introduction

Many physical systems of fundamental and industrial importance are significantly affected by the development of new materials, which is where materials engineering finds relevance. One of the main objectives of materials engineering is to establish process-structure-property relationship; the knowledge of which can be used to design tailor-made materials with desired properties. Experimental techniques and analytical strategies like first-principle calculations have been very popular. However, due to ever increasing demands of the industry in terms of the expected properties of materials, the experimental parameters to be analyzed to understand the property space kept exploding. This resulted in a multi-fold increase in the complexity of the system to be analyzed. With every additional parameter, *the curse of dimensionality* [84] states that the number of experiments required to understand this complex process-structure-property space increases exponentially. Additionally, huge amounts of data produced by these conventional experimental and analytical techniques along with high-throughput experimentation poses a great challenge for materials scientists.

However, increasing computational capabilities provide promising set of numerical strategies to performing large-scale data-mining thus converting this challenge to an opportunity. These computational strategies are aggregated under a single title called *Materials Informatics* [112]. *Materials Informatics* is a new branch of materials engineering that focuses on applying advanced information processing techniques to analyzing the data produced from high-throughput experimentation of materials. These combinatorial techniques are expected to profoundly reduce the (2-10 year) time required for a material discovered before being implemented in engineering practice [101]. This thesis addresses the challenges in analyzing huge datasets by leveraging the advanced mathe-

matical and computational techniques in order to establish process-structure-property relationship of materials.

1.2 Thesis Organization

This thesis is compiled based on the journal paper format, which means that each chapter (except Introduction and Conclusion) of the thesis is a manuscript published in, accepted by, submitted to, and/or prepared for submission to scholarly journals and proceedings (or modified from those versions). While chapter 1 provides a general introduction to the entire thesis, chapters 2, 3, and 4 constitute the body of the thesis, and finally chapter 5 provides a summary of the thesis.

Chapter 2 is a paper [119] to be submitted to the Computational Materials Science, 2013. This paper reviews various spectral based techniques that efficiently unravel linear and nonlinear structures in the data, which can subsequently be used to tractably investigate process-structure-property relationships. We compare and contrast the advantages and disadvantages of these techniques and discuss the mathematical and algorithmic underpinning of these methods. In addition, we describe techniques (based on graph-theoretic analysis) to estimate the optimal dimensionality of the low-dimensional parametric representation. We show how these techniques can be packaged into a modular, computationally scalable software framework with a graphical user interface - **S**calable **E**xtensible **T**oolkit for **Di**mensionality **R**eduction (SETDiR). This interface helps to separate out the mathematics and computational aspects from the material science applications, thus significantly enhancing utility to the materials science community. The applicability of this framework in constructing reduced order models of complicated materials data is illustrated with an example dataset of apatites described in structural descriptor space. Cluster analysis of the low-dimensional plots yielded interesting insights into correlation between several structural descriptors like ionic radius and covalence with characteristic properties like apatite stability. This information revealed crucial insights to promote the use of apatites as an antidote in lead poisoning.

Chapter 3 is a paper [121] to be submitted to the Scientific Programming, 2013. This chapter 3

extends the mathematics of [119] to illustrate a systematic analysis of spectral dimensionality reduction techniques and recast them into a unified view that can be exploited by dimensionality reduction algorithm designers. We subsequently identified the common computational building blocks required to implement a spectral dimensionality reduction method. We used this insight to design and implement a parallel framework for dimensionality reduction that can handle large datasets, and that scales to thousands of processors. We demonstrated the capability and scalability of this framework on several test data-sets. Additionally, we also showcased the applicability and potential of the framework towards unraveling complex process-structure relationships by studying the processing pathways of plastic solar cells.

Chapter 4 is a paper [120] which was submitted to the Computational Materials Science, 2013. This chapter 4 represents a graph-based formulation of the problem of precipitate characterization from point cloud Atom Probe Tomograph (APT) data. We present a robust, heuristic-free graph-theoretic methodology to solve the formulated problem and provide an implementation of it along with the results obtained by applying the **Gra**ph-theoretic methods to extract **P**recipitate **Top**ology framework (GraPTop) to three APT point cloud datasets of Al-Mg-Sc alloy. Our framework is robust due to its independence from heuristics like concentration level. We envision applying this framework to an array of datasets obtained from atom probe reconstruction where each dataset is prepared by regulated variation in the process of fabrication. This process of parametric study of the space can give interesting insights into the relationship between the topology of the precipitates and the fabrication process.

Chapter 5 provides a brief summary of the thesis and lists certain key achievements. This chapter also presents a compilation of open-ended ideas to guide the future course of action.

1.3 Literature Review

Experimental and analytical (or first-principle based calculations) [11, 4, 41, 66] remained as very popular strategies to establish process-structure-property relationship for a long period of time. A few of the most popular quantum mechanical methods among materials scientists are

Density functional theory [63, 105, 64] and Dynamic Mean-Field Theory [49]. Density functional theory is a quantum-mechanical modelling used to map structural properties of elements using electron density functional. Dynamic mean-field theory evaluates electronic structure of a given material by extending the mean-field theory to quantum mechanics. Though popular and long existent, analytical strategies are tedious and resource-intensive. Subsequently, there emerged other alternative, shortcut analytical methods to partially address the difficulty of complex and tedious nature of the system in hand. For example, ab initio pseudo-potential theory for metals [6, 57, 27, 86], used approximate values of potentials to compute certain properties of metals.

Recent advances in nano-technology, sensors, and automation devices have marked an era of powerful tool called High-Throughput Screening (HTS). High-Throughput Screening is a scientific methodology to perform experiments in bulk allowing synthesis, process and screening of millions of materials at a time. HTS is a popular methodology not just in materials [53, 71], but also in various other fields like drug discovery and genomics [138, 160] producing large amounts of data. These data repositories (also referred to as combinatorial libraries) are a rich source of information to establish process-structure-property relationship. However, there exist two challenges here: (a) Given the exploding amounts of process, structure, and property variables constituting the variable space, the number of experiments to be performed can prove to be over-whelming and expensive; and (b) Large amounts of data thus produced poses a huge challenge to materials scientists.

Materials Informatics [112] provides a comprehensive solution for both these challenges by putting in use the advanced information processing techniques: (a) To search through the variable space and identify interested parameter range for experimentation and (b) To leverage advanced mathematical and computational techniques to analyze materials data. Furthermore, materials informatics can form a very good supplementary strategy to provide insights and work hand-in-hand not just with the experimentation but also with analytical techniques. Statistical techniques are also a common solution to analyzing data from HTS: [19].

Current thesis is collection of generic scalable computational frameworks based on advanced mathematical tools that were built during the process of analyzing huge materials data to establish process-structure-property relationships. Though intended for materials data, these frameworks

are generic in nature and can be applied to analyzing data irrespective of the domain. This thesis also spins-off answers to certain common questions encountered while processing of such large data. For example: choice of mathematical model, choice of algorithm, and working around heuristics.

CHAPTER 2. NONLINEAR DIMENSIONALITY REDUCTION TECHNIQUES FOR APATITES

A paper yet to be submitted

S. Samudrala, P. V. Balachandran, S. Broderick, K. Rajan & B. Ganapathysubramanian

As a first author of this paper, I (S. Samudrala) developed the computational framework of dimensionality reduction. Analysis of the results by applying the techniques to a materials dataset of apatites was performed by P. V. Balachandran and S. Broderick under the supervision of K. Rajan.

2.1 Abstract

Materials Science research has witnessed an increasing use of data mining techniques in establishing structure-process-property relationships. Significant advances in high-throughput experiments and computational capability have resulted in the generation of huge amounts of data. Various statistical methods are currently employed to reduce the noise, redundancy, and the dimensionality of the data to make analysis more tractable. Popular methods for reduction (like Principal Component Analysis) assume a linear relationship between the input and output variables. Recent developments in nonlinear reduction (Neural Networks, Self-Organizing Maps), though successful, have computational issues associated with convergence and scalability. This paper reviews various spectral based techniques that efficiently unravel linear and nonlinear structures in the data which can subsequently be used to tractably investigate structure-property-process relationships. We compare and contrast the advantages and disadvantages of these techniques and discuss the

mathematical and algorithmic underpinning of these methods. In addition, we describe techniques (based on graph-theoretic analysis) to estimate the optimal dimensionality of the low-dimensional parametric representation. We show how these techniques can be packaged into a modular, and computationally scalable software framework with a graphical user interface - **S**calable **E**xtensible **T**oolkit for **Di**mensionality **R**eduction (SETDiR). This interface helps to separate out the mathematics and computational aspects from the material science applications, thus significantly enhancing utility to the materials science community. The applicability of this framework in constructing reduced order models of complicated materials dataset is illustrated with an example dataset of apatites described in structural descriptor space. Cluster analysis of the low-dimensional plots yielded interesting insights into correlation between several structural descriptors like ionic radius and covalence with characteristic properties like apatite stability. This information is crucial as it can promote the use of apatites as an antidote in lead poisoning.

2.2 Introduction

Using data mining techniques to probe and establish structure-process-property relationships has witnessed a growing interest owing to its ability to accelerate the process of tailoring materials by design. Before the advent of data mining techniques, scientists used a variety of empirical and diagrammatic techniques [111], like pettifor maps [100], or finite-element methods [141, 83, 157, 23, 87] to establish relationships between structure and mechanical properties. Pettifor maps, one of the earliest graphical representation techniques, is exceedingly efficient except that it requires a thorough understanding and intuition about the materials. Recent progress in computational capabilities has seen the advent of more complicated paradigms - so called virtual interrogation techniques - that span from first principle calculations to multi-scale models. These complex multi-physics and/or statistical techniques and simulations [91, 158] result in an integrated set of tools which can predict the relationships between chemical, microstructure and mechanical properties producing an exponentially large collection of data. Simultaneously, experimental methods - combinatorial materials synthesis [134, 94], high-throughput experimentation,

atom probe tomography- allow synthesis and screening of large number of materials while generating huge amounts of multivariate data. A key challenge is to efficiently probe the data to extract correlations between structures, process, and property. This data-explosion motivated the use of data-mining techniques in material science to explore, design and tailor materials and structures. A key stage in this process is to reduce the size of the data, while minimizing the loss of information during this data reduction. This process is called data-dimensionality reduction. By definition, Dimensionality Reduction (DR) is the process of reducing the dimensionality of the given set of (usually unordered) data points and extracting the low-dimensional (or parameter space) embedding with a desired property (for example: distance, topology, etc;) being preserved throughout the process. Examples for DR methods are Principal Component Analysis (PCA) [88], Isomap [136], Hessian Locally Linear Embedding (hLLE) [39], etc. Applying DR methods enables visualization of the high-dimensional data and also estimates the optimal number of dimensions required to represent the data without considerable loss of information.

Data dimensionality reduction is not a novel concept. Page [102] describes different techniques of data reduction and their applicability for establishing structure-property relationships. Statistical methods like PCA (Principal Component Analysis) [113], FA (Factor Analysis) [17] have been used on materials data generated by first-principles or experimental methods. However, dimensionality reduction techniques like PCA or Factor Analysis to establish structure-property relationships traditionally assume a linear relationship among the variables. This is often not strictly valid; the data usually lies on a nonlinear manifold (or surface). Nonlinear Dimensionality Reduction (NLDR) techniques can be applied to unravel the non-linear structure from unordered data. An example of such application for constructing a low-dimensional stochastic representation of property variations in random heterogenous media is shown in [47]. Another exciting application of data dimensionality reduction is in combination with quantum mechanics based calculations to predict structure [99, 29, 42]. For a more mathematical list of linear and nonlinear DR techniques, the interested reader can consult [84, 140].

In this paper, the theory and mathematics behind various linear and non-linear dimensionality reduction methods is explained. Algorithms are sketched and advantages and disadvantages of

methods are discussed. This paper also discusses and tackles questions pertinent to optimal dimensionality and model reduction process for different input parameters - like 'What is the optimal dimensionality of the manifold on which the data lies?', 'How well does the elbow in the scree plot reflect the optimal dimensionality?', etc. The mathematical aspects of NLDR are packaged into an easy-to-use software framework called Scalable Extensible Toolkit for Dimensionality Reduction (SETDiR) which (a) provides a user-friendly interface that successfully abstracts user from the mathematical intricacies, (b) allows for easy post-processing of the data, and (c) represents the data in a visual format and allows the user to store the output in '.JPG' format, thus making data more tractable and providing an intuitive understanding of the data. We conclude by applying the techniques discussed on a dataset of apatites [40, 147, 146, 93, 108] described using several structural descriptors. Apatites($A_4^I A_6^{II}(BO_4)_6X_2$) have the ability to accommodate numerous chemical substitutions and hence can be used in the process of detoxification. Section 2.3 describes the concepts, the algorithms of each DR method, dimensionality estimators and post-processing techniques in detail. The software framework, SETDiR, developed to apply DR techniques is described in Section 2.4. Section 2.5 discusses the interpretation of low-dimensional results obtained by applying SETDiR to the apatites dataset. Section 2.6 refers to supplementary literature and conclusions of this paper.

2.3 Materials and Methods

The problem of dimensionality reduction can be formulated as follows. Consider a set $X = \{x_0, x_1, \ldots, x_{n-1}\}$ of n points, where $x_i \in \mathbb{R}^D$, and $D \gg 1$. We are interested in finding a set $Y = \{y_0, y_1, \ldots, y_{n-1}\}$, such that $y_i \in \mathbb{R}^d$, $d \ll D$ and $\forall_{i,j}\ |x_i - x_j|_h = |y_i - y_j|_h$. Here, $|a-b|_h$ denotes a specific norm that captures properties we want to preserve during dimensionality reduction [84]. For instance, by defining h as Euclidean norm we preserve Euclidean distance, thus obtaining a reduction equivalent to the standard technique of Principal Component Analysis (PCA) [88]. Similarly, defining h to be the angular distance (or conformal distance [14]) results in Locally Linear Embedding(LLE) [115] that preserves local angles between points. In a typical application [45, 153]

x_i represents a state of the analyzed system, e.g. temperature field, concentration distribution, etc. Such state description can be derived from the experimental sensor data or can be result of a numerical simulation. However, irrespective of the source, it is characterized by high dimensionality, that is D is typically of the order of 10^6 [151, 55]. While x_i represents just a single state of the system, common data acquisition setups deliver large collections of such observations, which correspond to the temporal or parametric evolution of the system [45]. Thus, the cardinality n of the resulting set X is usually large ($n \sim 10^4$–10^5). Intuitively, information obfuscation increases with the data dimensionality. Therefore, in the process of Dimensionality Reduction (DR) we seek as small a dimension d as possible, given the constraints induced by the norm $|a - b|_h$ [84]. Routinely, $d < 4$ as it permits, for instance, visualization of the set Y.

The key idea underpinning DR can be explained as follows. We encode desired information about X, i.e. topology or distance, in its entirety by considering all pairs of points in X. This encoding is represented as a matrix $A_{n\times n}$. Next, we subject matrix A to unitary transformation V , i.e. transformation that preserves norm of A, to obtain its sparsest form Λ, where $A = V\Lambda V^T$. Here, $\Lambda_{n\times n}$ is a diagonal matrix with rapidly diminishing entries. As a result, it is sufficient to consider only d entries of Λ to capture all the information encoded in A. These d entries constitute the set Y . The above procedure hinges on the fact that unitary transformations preserve original properties of A [51]. Note also, that it requires a method to construct matrix A in the first place. Indeed, what differentiates various spectral methods is the way information is encoded in A.

2.3.1 Dimensionality Reduction: Basic Ideas and Taxonomy

Before going further into the details of the functionality of DR methods, a brief taxonomy of these techniques is useful. A classification of DR methods can be carried out in various ways. Based on the topology of the manifold on which the data lies, they can be classified as:

- **Linear DR methods:** Linear DR methods assume that the dataset lies on a linear manifold. They are an efficient technique when the manifold is linear, but fail to retrieve the hidden structure if the manifold is nonlinear.

Eg: PCA, Multi-Dimensional Scaling (MDS) [81]

- **Nonlinear DR methods:** Nonlinear methods do not assume anything about the linearity of the manifold. Hence, they can extract the underlying structure of the manifold irrespective of whether the manifold is linear or nonlinear in the embedding space.
  Eg: Isomap, Locally Linear Embedding (LLE), Hessian LLE (hLLE)

Based on the property they preserve, the DR methods can be classified as:

- **Isometric DR methods:** They preserve pair-wise distances among all the input vectors in the given dataset.
  Eg: PCA, Isomap

- **Topology Preserving DR methods:**These methods preserve the topology or connectivity of the dataset. These methods tend to stretch or twist but do not tear the manifold.
  Eg: LLE, hLLE, Laplacian Eigenmaps (LE) [13]

2.3.2 Dimensionality Reduction Methods: Algorithms

We focus on four different DR methods: (a) Principal Component Analysis (PCA), a linear DR method; (b) Isomap, a non-linear isometry preserving DR method; (c) Locally Linear Embedding (LLE), a non-linear conformal preserving DR method; and (d) Hessian LLE (hLLE), a topology preserving DR method.

2.3.2.1 PCA: Principal Component Analysis

Principal Component Analysis (PCA) is a powerful and a popular DR strategy due to its simplicity and ease in implementation. It is based on the premise that the high dimensional data is a linear combination of a hidden low-dimensional axes. PCA then extracts the latent parameters or low-dimensional axes by reorienting the axes of the high-dimensional space in such a way that the variance of the variables is maximized [84].

PCA Algorithm:

1. Compute the pair-wise euclidean distance matrix $[E]$ from the input matrix $[X]$.

2. Construct a matrix $[W^*]$ such that the elements of $[W^*]$ are square of the elements of the euclidean distance matrix $[E]$.

3. Find the dissimilarity matrix $[A]$ by double centering $[W^*]$: $[A] = [H^T][W^*][H]$

$$H_{ij} = \begin{cases} (1 - 1/n) \; \forall \;\; \mathbf{i} = \mathbf{j}, \\ (-1/n) \; \forall \;\; \mathbf{i} \neq \mathbf{j}. \end{cases} \tag{2.1}$$

4. Solve for the largest d eigen-pairs of A: $[A] = [U][\Lambda][U^T]$

5. Construct the low-dimensional representation in $\mathbb{R}^d$ from the eigen-pairs: $[Y] = [I] * [\Lambda]^{1/2} * [U^T]$.

The limitation of PCA lies in its assumption that the data lies on a linear space, and hence performs poorly on data that lie on a nonlinear manifold. In these cases, PCA also tends to over-estimate the dimensionality of the data.

2.3.2.2 Isomap

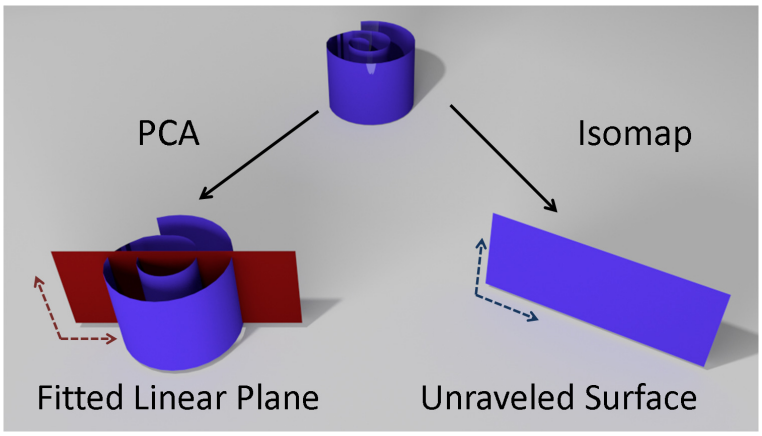

Figure 2.1 Comparison of performance of PCA and Isomap on a dataset lying on nonlinear manifold

Isomap [136] relaxes the assumption of PCA that the data lies on a linear space. A classic example of a non-linear manifold is the swiss roll. Figure 2.1 shows how PCA tries to fit best

linear plane while Isomap unravels the low-dimensional surface. Isomap essentially smooths out the non-linear manifold into a corresponding linear space and subsequently applies PCA. This smoothing out can intuitively be understood in the context of the spiral, where the ends of the spiral are pulled out to straighten the spiral into a straight line. Isomap accomplishes this objective mathematically by ensuring that the geodesic distance between data-points are preserved under transformations. The geodesic distance is the distance measured along the curved surface on which the points rest [84]. Since it preserves (geodesic) distances, Isomap is an isometry (distance-preserving) transformation. The underlying mathematics of the Isomap algorithm assumes that the data lies on a manifold which is convex (but not necessarily linear). Note that both PCA and Isomap are isometric mappings; PCA preserves pair-wise Euclidean distances of the points, while Isomap preserves the geodesic distance.

Isomap Algorithm:

1. Compute the pair-wise euclidean distance matrix $[E]$ from the input matrix $[X]$.
2. Compute the k-nearest neighbors from the distance matrix $[E]$.
3. Compute the pair-wise geodesic distance matrix $[G]$ from $[E]$.
4. Construct a matrix $[W^*]$ such that the elements of $[W^*]$ are square of the elements of the geodesic distance matrix $[G]$.
5. Find the dissimilarity matrix $[A]$ by double centering $[W^*]$: $[A] = [H^T][W^*][H]$

$$H_{ij} = \begin{cases} (1 - 1/n) \; \forall \;\; \mathbf{i} = \mathbf{j}, \\ (-1/n) \; \forall \;\; \mathbf{i} \neq \mathbf{j}. \end{cases} \tag{2.2}$$

6. Solve for the largest d eigen-pairs of A: $[A] = [U][\Lambda][U^T]$
7. Construct the low-dimensional representation in $\mathbb{R}^d$ from the eigen-pairs: $[Y] = [I] * [\Lambda]^{1/2} * [U^T]$.

The non-linearity in the data is accounted for by using geodesic distance metric. Since computation of geodesic distance without the knowledge of the low-dimensional surface is close to

impossible, graph distance is used to approximate the geodesic distance [15]. Graph distance between a pair of points in a graph (V, E) is the shortest path connecting the two given points. The graph distances are calculated for a given graph using Floyd's algorithm [44].

2.3.2.3 Locally Linear Embedding

In contrast to PCA and Isomap methods which preserve distances, Locally Linear Embedding (LLE) [115] preserves the local topology (or local orientation, or angles between data-point). LLE method uses the notion that locally the (non-linear) manifold on which the data lie on is well-approximated by a d-dimensional Euclidean space($\mathbb{R}^d$). In other words, the manifold is locally linear. The algorithm first divides the manifold into patches, reconstructs each point in the patch based on the information (or weights) obtained from its neighbors (i.e. infer how a specific point is located with respect to its neighbors). This process extracts the local topology of the data. Finally, the algorithm reconstructs the global structure by combining individual patches and finding an optimized, low-dimensional representation. Numerically, local topology information is constructed by finding the k-nearest neighbors of each data point and reconstructing each point from the information about the weights of the neighbors. The global reconstruction from the local patches is accomplished by assimilating the individual weight matrices to form a global weight matrix $[W]$ and evaluating the eigenvalues of normalized global weight matrix $[A]$.

LLE Algorithm:

1. For (i=1:n) each of the n input vectors from $X = \{x_0, x_1, \ldots, x_{n-1}\}$ of n points, where $x_i \in \mathbb{R}^D$:

   (a) Find the k nearest neighbors of the input vector x_i.

   (b) Construct the local covariance or Gram matrix $G(i)$

   $$g_{r,s}(i) = (x_i - \nu(r))^T (x_i - \nu(s))$$

   where $\nu(r)$ and $\nu(s)$ are neighbors of x_i.

   (c) Weights can be computed by solving for the linear system:$[G(i)].w(i) = 1$ where 1 is a $k \times 1$ vector of ones.

2. Knowing the vectors $w(i)$, build the sparse matrix W such that for each ith row $W(i,j) = 0$ if x_i and x_j are not neighbors and the corresponding linear coefficient obtained by solving the linear equation $[G(i)].w(i) = 1$ if otherwise.

3. From W build A: $[A] = (I - W)^T(I - W)$

4. Compute the eigenpairs for A: $[A] = [U][\Lambda][U^T]$

5. Compute the low-dimensional points in $\mathbb{R}^d$ from the eigen-pairs: $[Y] = n^{0.5} * [U]$

2.3.2.4 Hessian LLE

Hessian Locally Linear Embedding (Hessian LLE or hLLE) [39] is an improvement upon LLE and Laplacian Eigenmaps [13], which replaces the Laplacian (first derivative) operator with a Hessian (second derivative) operator over the given connected graph. hLLE constructs patches, performs a local PCA on each patch, constructs a global Hessian from the eigenvectors thus obtained and finally finds the low dimensional representation from the eigenpairs of the Hessian. hLLE (Hessian Locally Linear Embedding) is a topology preservation method and assumes that the manifold is locally linear.

hLLE Algorithm:

1. At each given point x_i , construct a k x n Neighborhood matrix $[M_i]$ such that each row of the matrix represents a point

$$x_j = x_j - \bar{x_i},$$

where $j\epsilon[0, N)$ and x_i is the mean of the k neighboring points.

2. Perform singular value decomposition of the constructed $[M_i]$ to obtain [U], [V], [D].

3. Construct the $(N * d(d+1)/2)$ local hessian matrix $[X]^i$ such that the first column is a vector of all ones and the next d columns are the columns of U followed by the products of all the d columns of $[U]$.

4. Compute Gram-Schmidt orthogonalization [51] on the local hessians $[X]^i$ and assimilate last $d(d+1)/2$ orthonormal vectors of each to construct the global hessian matrix $[A]$ [39].

5. Compute the eigenpairs of the hessian matrix:$[A] = [W][\Lambda][W]^T$

6. Compute the low-dimensional points $[Y]$ in $\mathbb{R}^d$ from the eigenpairs: $[Y] = [W] * ([W]^T * [W])^{-1/2}$

An important point to note here is that as discussed in section 2.3, matrix $[A]$ encodes the required information for each of the DR techniques and the construction of this matrix is what differentiates a spectral DR method from the rest. Matrix $[A]$ is a normalized euclidean matrix in the case of PCA, a normalized geodesic matrix in the case of Isomap, a normalized Hessian matrix for hLLE, and so on.

2.3.3 Dimensionality Estimators

A key step in constructing the low-dimensional points from the data is the choice of the low-dimensionality or optimal dimensionality d. Methods like PCA and Isomap have an implicit technique to estimate the low-dimensionality (approximately) using scree plots. We introduce a graph-based technique that rigorously estimates the latent dimensionality of the data, that can be used in conjunction with the scree-plot.

2.3.3.1 Dimensionality from the scree plot

Scree plot is a plot of the eigenvalues with the eigenvalues arranged in decreasing order of their magnitude. Scree plots obtained from PCA and Isomap (distance preserving methods) give an estimate of the dimensionality. A heuristic method of identifying the dimensionality by identifying the elbow in the scree plot. A more quantitative estimate of dimensionality is estimated by choosing a value for δ that ensures a threshold of the minimum percentage variability. If $\lambda_1 > \lambda_2 > ..\lambda_n$ are the individual eigenvalues arranged in descending order, the percentage variability $(p_{var}(d))$

covered by considering first d eigenvalues is given by:

$$p_{var}(d) = 100 * \frac{\sum_{i=1}^{d} \lambda_i}{\sum_{i=1}^{n} \lambda_i} \leq \delta$$

2.3.3.2 Geodesic Minimal Spanning Tree Estimator (GMST Estimator):

A tree is a graph where each pair of vertices is connected exactly with one path. A spanning tree of a graph G(V,E) is a sub-graph that traces all the vertices in the graph. A Minimal Spanning Tree (MST) of a weighted graph G(V,E,W) is a spanning tree with optimal sum of the edge weights (length of the MST) along the tree. A Geodesic Minimal Spanning Tree (GMST) is an MST with edge weight representing geodesic distance.

Computationally, GMST is computed using Prim's (greedy) algorithm [109]. Starting with any random vertex in the weighted graph G(V,E,W), Prim's algorithm constructs an MST by picking edges that have (a) minimal weights and (b) that connect to an untraced vertex. By joining one edge after the other following these constraints, at the end of algorithm, one has a tree that spans all the n vertices with $n-1$ edges and whose sum of edge weights is optimal.

We have recently utilized a dimensionality estimator based on Breadwood-Halton-Hammersley Theorem (BHH theorem) [12]. This theorem states that the rate of convergence of the length of minimal spanning tree (L_n) gives us a measure of the dimensionality d as $n \to \infty$. This allows one to express the dimensionality (d) of an unordered dataset as a function of the length of GMST of the graph. Specifically, the slope of a $\log(n)$ vs. $\log(L_n)$ plot constructed by calculating the GMST with respect to increasing size of data-points (n) provides an estimate of the dimensionality: $d = \frac{1}{(1-m)}$, where m is the slope of the log-log plot.

2.3.3.3 Correlation Dimension:

Correlation dimension is given by: $d_{cor}(\epsilon_1, \epsilon_2) = \frac{log(\hat{C}_2(\epsilon_2)) - log(\hat{C}_2(\epsilon_1))}{log(\epsilon_2) - log(\epsilon_1)}$ where $\hat{C}_2(\epsilon_2)$ is a measure of proportion of distances less than ϵ [52, 84]. Intuitively, these epsilon values are like window ranges through which one zooms through the data. This means if they are too small the data

would look like individual points and if too huge, the entire dataset is seen as a single fuzzy spot. Hence, correlation dimension is sensitive to the epsilon values. A qualitative technique to choose epsilon is adopted to estimate the dimensionality of the data. This is done by plotting a graph between $\hat{C}_2(\epsilon_2)$ and ϵ and choosing a range of ϵ where the graph is relatively stable. However, one important point to note is that the correlation dimension provides the user with a lower bound of the optimal-dimensionality.

2.3.4 Post-Processing:Clustering

Clustering is the post-processing step of the low-dimensional plots obtained by applying dimensionality reduction. Clustering the obtained low-dimensional points often helps in extracting interesting features and thus allows one to draw conclusions that can provide insights into the physics that drives the data. Clustering also has the capability to quantify the intuitive ideas generated by visual analysis of the plots. Clustering of unordered set of points is a common and well-studied problem [156, 20, 58]. For a start, we implemented k-means clustering (iterative) algorithm [58] that is described as follows:

1. Read input $[Y]_{n\times d}$ and the number of clusters κ.
2. Initialize the κ-centroids to a set of κ random points in $[Y]$ and store them in $[cent]_{\kappa\times 1}$.
3. Initialize the cluster index as an array of -1s in $[indx]_{n\times 1}$.
4. For each low-dimensional point $y_i \epsilon [Y]$:
   (a) Compute the distance of y_i from each of the κ centroid points and store them in a vector $[D_i]_{\kappa\times 1}$.
   (b) Compute the index of occurrence of the minimum in the vector $[D_i]$.
   (c) Update the cluster index of the current point y_i in $indx_i$.
5. Update old centroid as: $[oldcent] = [cent]$.
6. Compute the centroids of the newly formed clusters and store them in $[cent]$

7. For each centroid point: $i = 1 : \kappa$

   (a) if($cent_i == oldcent_i$) continue;

   (b) Else go to 5

8. Output is the cluster mapping: ($[Y]$,$indx$).

The k-means clustering requires the user to input a heuristic κ of number of clusters. However, since the low-dimensional points $[Y]$ are easy to visualize to estimate a suitable κ value should not be difficult.

2.4 Software: SETDiR

These DR techniques can be packaged into a modular, scalable framework for ease of use by the materials science community. We call this package, Scalable Extensible Toolkit for Dimensionality Reduction (SETDiR). This framework contains two major components:

1. Core functionality: Developed using C++

2. User Interface: Developed based on Java (Swings)

Fig. 2.2 describes the scope of the functionality of both modules in SETDiR. Details of the implementation are described in the subsequent subsections.

2.4.1 Core Functionality

Functionality is developed using Object Oriented C++ programming language. It implements the following methods: PCA, Isomap, LLE and dimensionality estimators like: GMST and correlation dimension estimators [84].

2.4.2 User Interface

A graphical user interface (shown in fig. 2.3) is developed using Java^{TM} Swings Components with the following features which make it user-friendly:

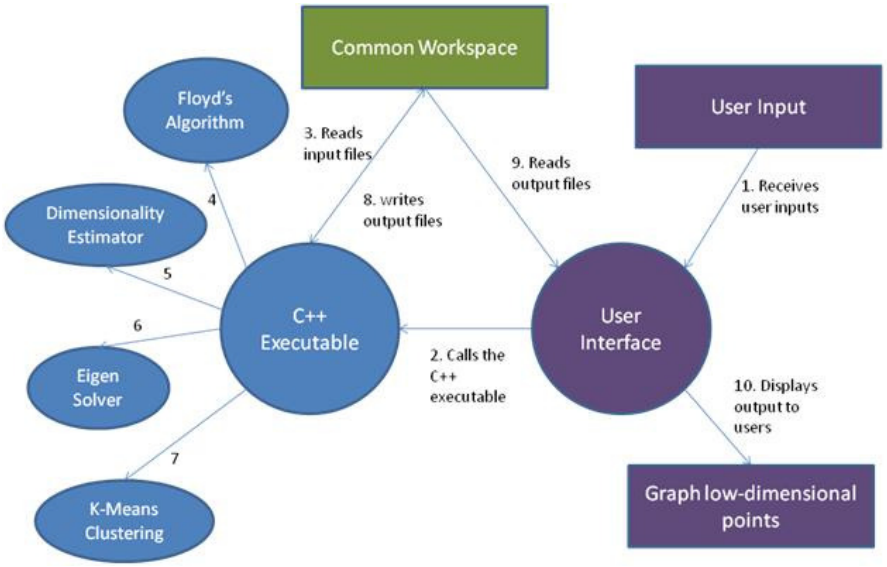

Figure 2.2 Description of the software

1. Abstracts the user from the mathematical and programming details.

2. Displays the results graphically and enhances the visualization of low-dimensional points.

3. Easy Post-processing of Results: In-built cluster analysis, ability to save plots as image files.

4. Organized settings tabs: Based on the niche of the user, the solver settings are organized as: Basic User and Advanced User tabs which abstract a new or a naive user from, otherwise overwhelming, details.

This framework can be downloaded from SETDiR Download [1]. We next showcase the framework and the mathematical strategies on apatites dataset.

2.5 Results and Discussion

In this section of the paper, we compare and contrast the algorithms on an interesting dataset of apatites with immense technological and scientific significance. Data dimensionality reduction offers unique insights into the originally intractable datasets by enabling visual clustering and pattern association. Apatites have the ability to accommodate numerous chemical substitutions and

[1] http://setdir.engineering.iastate.edu/doku.php?id=download

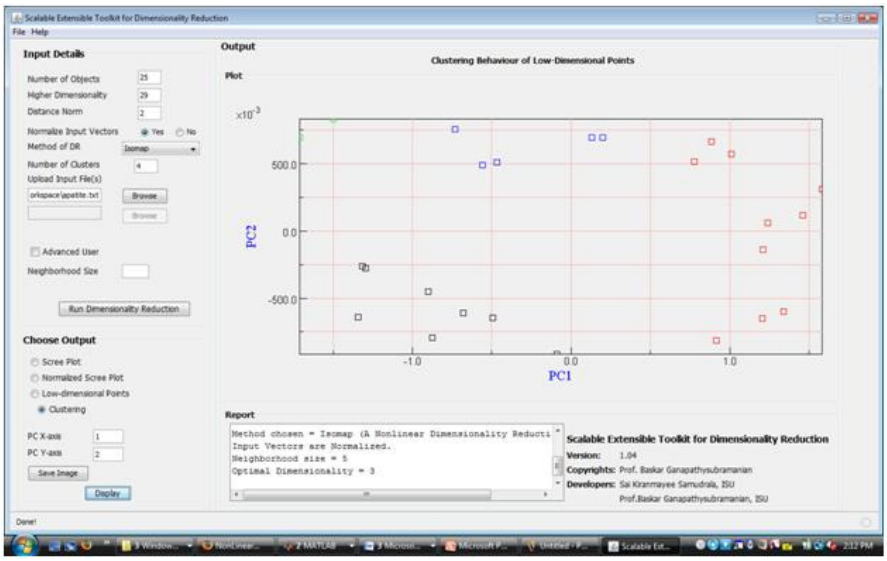

Figure 2.3 Snapshot of clustering pattern displayed using SETDiR for apatite dataset

exhibit a broad range of multifunctional properties. The rich chemical and structural diversity provides a fertile ground for the synthesis of technologically relevant compounds [40, 147, 146, 93, 108]. Chemically apatites are conveniently described by the general formula $A_4^I A_6^{II}(BO_4)_6X_2$, where A_I and A_{II} are distinct crystallographic sites that usually accommodate larger monovalent (Na^+, Li^+, etc.), divalent ($Ca^{2+}, Sr^{2+}, Ba^{2+}, Pb^{2+}, etc.$) and trivalent ($Y^{3+}, Ce^{3+}, La^{3+}$, etc.) and the X-site is occupied by halides (F^-, Cl^-, Br^-), oxides and hydroxides. Establishing the relationship between the microscopic properties of apatite complexes with those of the macroscopic properties can help us in gaining understanding and promoting the use of apatites in various daily life applications. For example, information about the relative stability of the apatite complexes can promote the utilization of apatites as an antidote for lead poisoning (by finding an apatite which is more stable than a lead apatite). DR techniques can be used to establish structure-property relationship for apatites described using various structural descriptor by enhancing the visualization and understanding of the data.

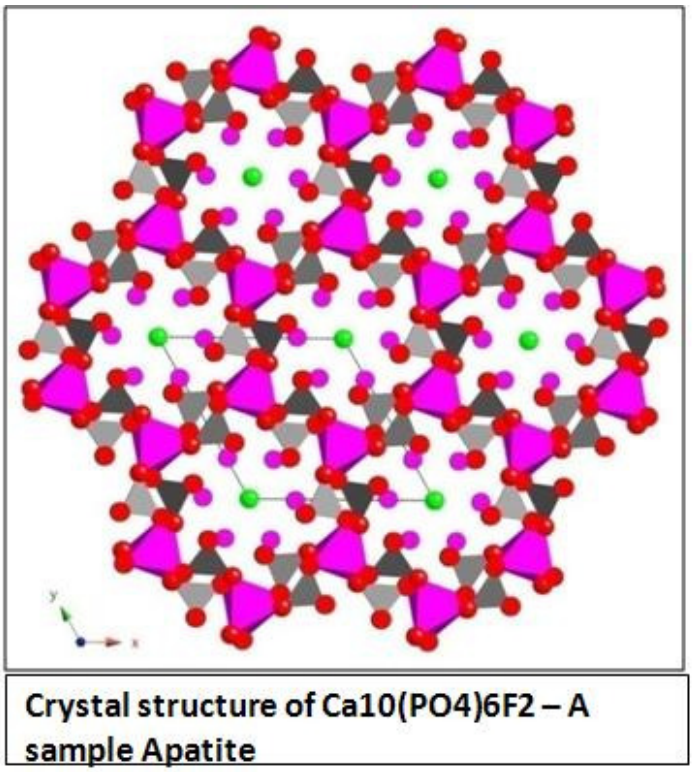

Figure 2.4 Crystal structure of a typical $P6_3/mCa^I_4Ca^{II}_6(PO_4)_6F_2$ apatite with hexagonal unit cell. [8, 7]

2.5.1 Apatite Data Description

The crystal structure of a typical $P6_3/mCa^I_4Ca^{II}_6(PO_4)_6F_2$ apatite with hexagonal unit cell is shown in the fig. 2.4 with the atoms projected along the (001) axis.The polyhedral representation of A^IO_6 and BO_4 structural units are clearly shown with the Ca^{II}-site (pink atoms) and F-site (green atoms) occupying the tunnel. Thin black line represents the unit-cell of the hexagonal lattice.

The sample apatite dataset considered consists of 25 different compositions described using 29 structural descriptors. These structural descriptors, when modified, affect the crystal structure of the composition [8]. By establishing the relationship between the crystal structure and these structural descriptors and analyzing the clustering of different compositions, conclusions can be drawn about how the changes in these structural descriptors (defining the microscopic properties) of the crystal structure can affect the macroscopic features (like melting point, Young's modulus, etc.). The bond length, bond angle, lattice constants and total energy data is taken from the

work of Mercier et al. [93], the ionic radii data is taken from the work of Shannon [126] and the electronegativity data is based on the Pauling's scale [106]. The ionic radii of A^I-site (rAI) has a coordination number nine, rA^{II} has a coordination number seven (when the X-site is F^-) or eight (when the X-site is Cl^- or Br^-). Our database describes Ca, Ba, Sr, Pb, Hg, Zn and Cd in the A-site, P, As, Cr, V and Mn in the B-site and F, Cl and Br in the X-site. The twenty-five compounds considered in this study belong to $P6_3/m$ hexagonal space group. We utilize SETDiR on the apatite data and present some of the results below.

2.5.2 Dimensionality Estimation

SETDiR first estimates the dimensionality using the Scree Plot. A Scree Plot is a plot of eigenvalue indices vs eigenvalues. The occurrence of an elbow (or a sharp drop in eigenvalues) in a scree plot gives the estimate of the dimensionality of the data. Fig. 2.5 displays the scree plots when the input vectors $\{x_0, x_1, \ldots, x_{n-1}\}$ were normalized with respect to that when they were not normalized. For comparison, we plot the eigenvalues that are obtained from both PCA and Isomap. This plot shows how the second eigenvalue collapses to zero when the input vectors are not normalized and hence emphazises the importance of normalization of input vectors. It is also interesting to compare the eigenvalues of PCA and Isomap for normalized input. PCA being a linear method over-estimates the dimensionality as 5, while Isomap estimates it to be 3. SETDiR subsequently uses the Geodesic Minimal Spanning Tree method to estimate the dimensionality of the apatite data. This method gives a rigorous estimate of 3, which matches the outcome of the more heuristic Scree Plot estimate.

2.5.3 Low-dimensional Plots

Fig. 2.6(Left) shows the 2D plot between principal components 2 and 3. The reason for showing this classification map is that PC2-PC3 map captures pattern that is similar to IsoMap components 1 and 2. While we find associations among compounds that are exactly the same as shown in Fig. 2.6(Right), the nature of information is manifested in different ways. This is mainly attributed to the difference in the governing mathematics of the two techniques, where PCA is essentially a

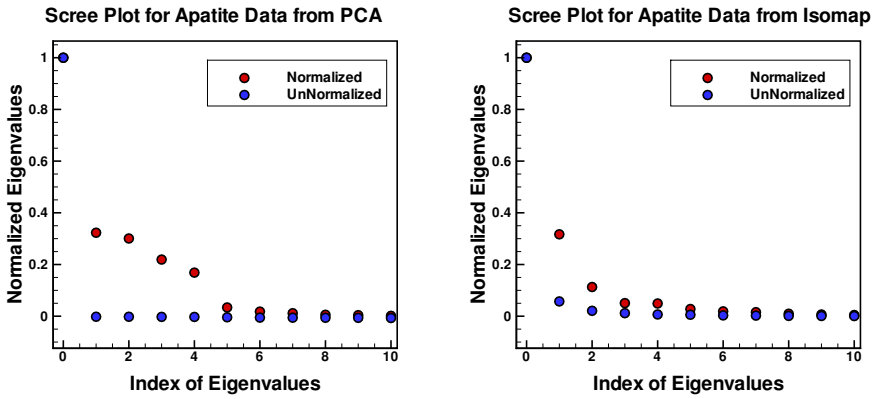

Figure 2.5 Scree plots for PCA and Isomap -Normalized vs. Unnormalized input [9]

linear technique and IsoMap is a non-linear technique. To further interpret the hidden information captured by IsoMap classification map (Fig. 2.6), we have focused on the three regions separately.

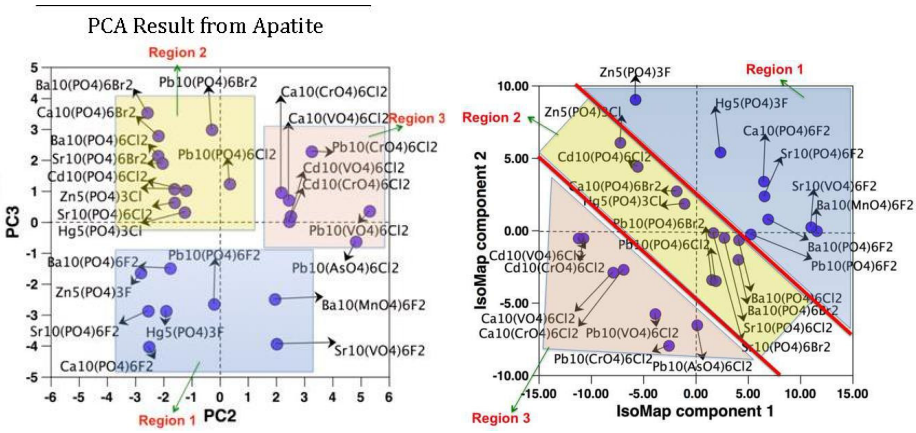

Figure 2.6 Apatite PCA (left) and Isomap (right) Result Interpretation [7]

Fig. 2.6 (Right) shows a two-dimensional classification map with IsoMap components 1 and 2 in the orthogonal axes [9]. The two-dimensional classification map groups various apatite compounds into three distinct regions that capture various interactions between A, B, and X-site ions in complex apatite crystal structure. Region 1 corresponds to apatite compounds with fluoride (F) ion in the X-site. All apatite compounds in this region contain only F in the X-site, but has different Asite (Ca, Sr, Pb, Ba, Cd, Zn) and B-site elements (P, Mn, V). Therefore, this unique region classifies F-apatites from Cl and Br-apatites. Region 2 belongs to apatite compounds with phosphorus (P) ion in the B-site and contains Cl and Br ions in the X-site. The uniqueness of this region is manifested

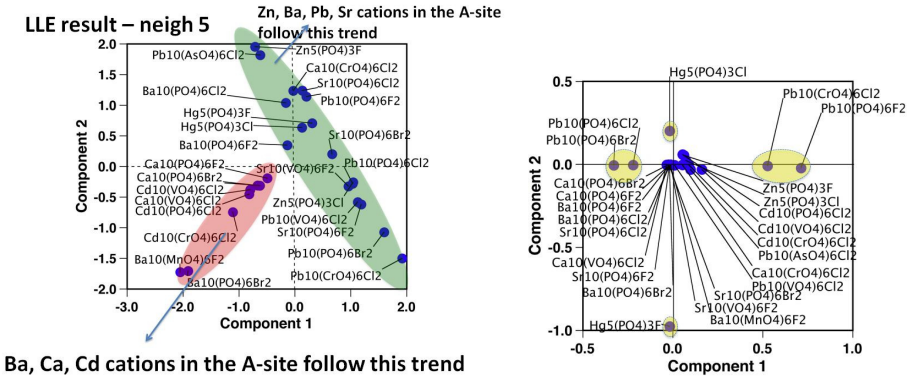

Figure 2.7 Apatite LLE (left) and hLLE (right) Result Interpretation. [7]

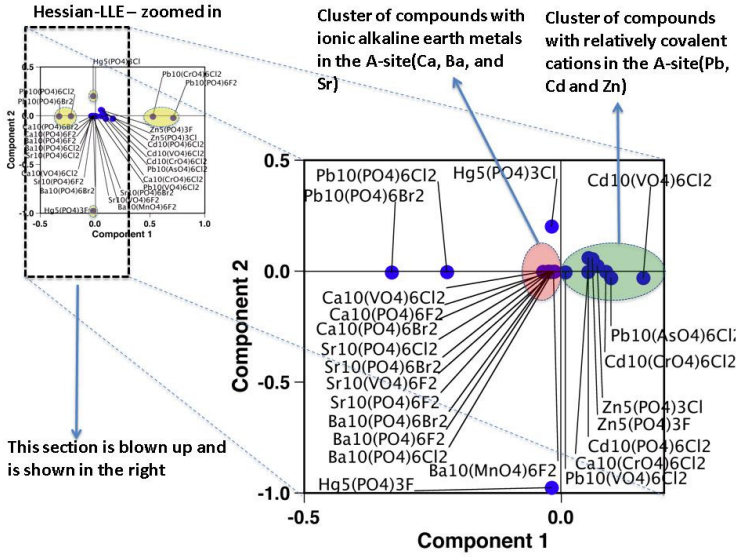

Figure 2.8 Apatite hLLE Result Interpretation [7]

mainly due to the presence of only smaller P ions in the B-site. Similarly, region 3 belongs to apatite compounds with Cl ions in the X-site and contains larger B-site Cr, V and As cations.

Fig. 2.7 (Right) presents the results from hLLE. It can be observed that the compounds that have highly covalent A-site cation (e.g. Hg^{2+} and Pb^{2+}) and highly covalent B-site cation (P^{5+}) clearly separate out from the rest. An exception to this rule is $Pb_{10}(CrO_4)_6Cl_2$ and the physical reason behind this could be attributed to the tetrahedral distortion of Cr^{5+} cation, which causes structural distortion. For example, $Sr_{10}(PO_4)_6Cl_2$ has a $P6_3/m$ hexagonal symmetry whereas

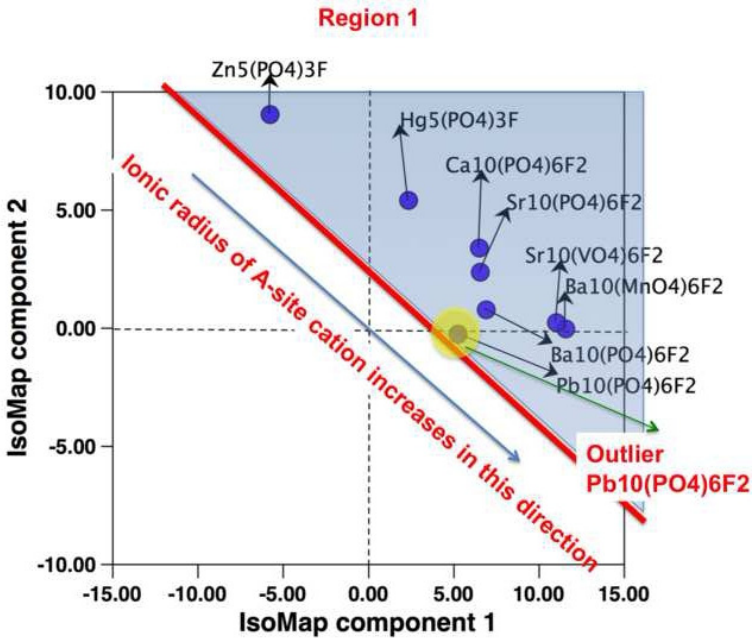

Figure 2.9 Apatite Isomap Result Interpretation Region 1 [7]

$Sr_{10}(CrO_4)_6Cl_2$ has a reduced P63 symmetry [146]. This result is in agreement with our PCA-derived structure map work, where we find that $Pb_{10}(CrO_4)_6Cl_2$ does not obey the general trend and is seen as an exception. Based on our PCA work [9], we attributed the cause for this exception to two bond distortion angles: rotation angle of $A^{II} - A^{II} - A^{II}$ triangular units and the angle that bond $A^I - O_1$ makes with the c-axis.

Fig. 2.8 shows a zoomed in plot of Hessian LLE result [2]. Around the origin we can find two clusters of compounds: (a) one on the left and have negative component 1 values correspond to compounds that have ionic alkaline earth metal cations in the A-site and (b) one on the right with positive component 1 value correspond to compounds that have covalent A-site cations. An exception here is $Ca_{10}(CrO_4)_6Cl_2$ found among the covalent A-site cluster and the physical reason behind this could be attributed to the tetrahedral Cr^- cation. This result suggests that $Ca_{10}(CrO_4)_6Cl_2$ may have a reduced symmetry. Further experimental and theoretical calculations are required to validate this findings. PCA did not capture this trend and this is a new finding. The physical

[2]Hessian LLE is highly sensitive to neighborhood size and is much more sensitive to the input estimated dimensionality. Incorrect input of estimated dimensionality implies construction of tangent planes of incorrect dimensions which, in turn, implies sub-optimal low-dimensional representation.

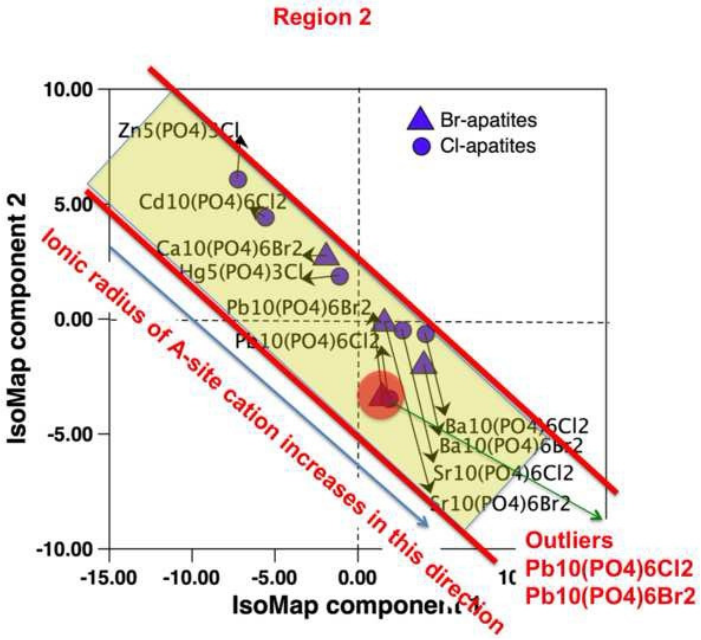

Figure 2.10 Apatite Isomap Result Interpretation Region 2 [7]

reason for $Pb_{10}(AsO_4)_6Cl_2$ and $Pb_{10}(VO_4)_6Cl_2$ to cluster around origin (0,0) could be attributed to the large ionic size of V^{5+} and As^{5+} cations.

In Fig. 2.9, the ionic radius of A-site elements increases along the direction, with Zn^{2+} cation being the smallest and Ba^{2+} being the largest. This ionic radii trend is not very clear in the PC2-PC3 classification map (Fig. 2.6). Besides the ionic radii trend captured using IsoMap, $Pb_{10}(PO_4)_6F_2$ apatite is identified as an outlier. Ionic size of Pb^{2+} is larger than Ca^{2+} but smaller than Sr^{2+} cation. Ideally, $Pb_{10}(PO_4)_6F_2$ should have been between $Ca_{10}(PO_4)_6F_2$ and $Sr_{10}(PO_4)_6F_2$ compounds in the map. However, this was not the case. The physical reason behind this observation could be manifested in the electronic structure of Pb^{2+} ions [90]. The theoretical electronic structure calculations indicate that in the partial density of states curves, the Pb^{2+} ions have active 6s2 lone-pair electrons that hybridize with oxygen 2p electrons resulting in a strong covalent bond formation. This feature was identified to be unique with respect to Pb^{2+} ions and this caused the Pb-apatites to behave differently. In our database, the electronic structure information of A-site elements was quantified using Pauling's electronegativity data. While PCA captures this

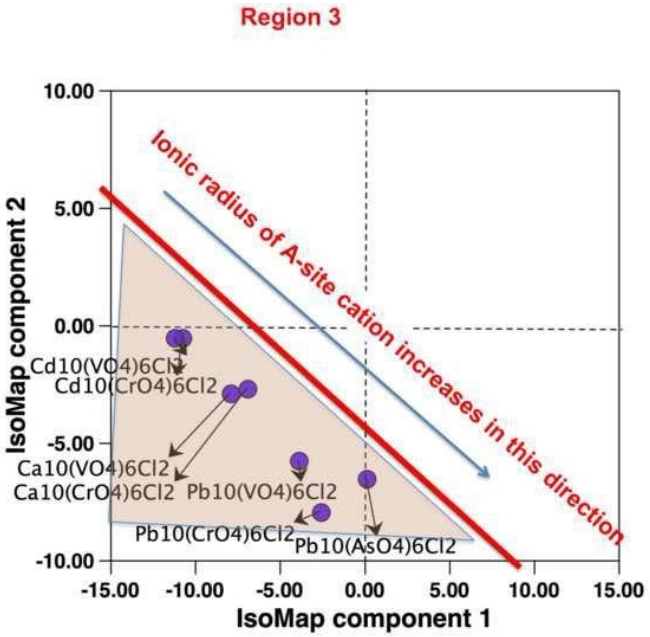

Figure 2.11 Apatite Isomap Result Interpretation Region 3 [7]

behavior, the dominating effect of the electronic structure of Pb^{2+} ions is more transparent within the mathematical framework of IsoMap analysis. Besides, from Fig. 2.9 it can also be inferred that the bond distortions of Zn-apatite is different from other compounds. This trend correlates well with the non-existence of $Zn_{10}(PO_4)_6F_2$ compounds due to the difficulty in experimental synthesis [43]. On the other hand, the relative correlation position of $Hg_{10}(PO_4)_6F_2$ compound indicate that it might be difficult to experimentally synthesize fully stoichiometric $Hg_{10}(PO_4)_6F_2$, but partial substitution in the host lattice of apatite compounds with Ca, Sr, Pb or Ba in the A-site might be a feasible practical solution. The ionic size of Hg^{2+} is very close to that of Ca^{2+} and if cation size is the key factor that governs the apatite stability, from Fig. 2.10 the bond distortions of Hg and Ca compounds are closely correlated. When this structural association between Hg and Ca compounds is combined with the low energy-cost of Ca-apatites, we conclude that Ca-apatites could be tailored to immobilize toxic Hg element. In Fig. 2.10, the region 2 alone is highlighted where we find a clear trend of apatite compounds with respect to the ionic radii of A-site elements. Similar to region 1, Pb-apatites manifest themselves as outliers in region 2. The unique electronic

structure of Pb^{2+} cations in forming a covalent bond with oxygen 2p electrons is identified as the reason for the deviation of Pb-apatites from the expected trend. The covalent chemical bonding among Pb-compounds appears to be independent of X-site anion, when the B-site is occupied by phosphorus cations. In Fig. 2.10, $Hg_{10}(PO_4)_6Cl_2$ compound is found to be closely associated with $Ca_{10}(PO_4)_6Br_2$ indicating some similarity in the bond distortions of the two compounds. In comparing the relative correlation position of all Cl-containing apatites (except Pb-based compounds) in region 2, the bond distortions in $Hg_{10}(PO_4)_6Cl_2$ appear to favour stable apatite compound formation.

Fig. 2.11 describes region 3 where we find clusters of apatite compounds with Cl ions in the X-site and contain larger V, Cr and As cations in the B-site. The ionic radius of A-site element increases in the direction as shown in the figure and in this case, the Pb-apatites are not outliers. The presence of large V, Cr and As cations (compared to smaller P cations in region 1 and 2) in the B-site were identified as the reason for this behaviour. Besides, region 3 also identifies the existence of complex relationship observed in Cl-apatites with Pb in the A-site and containing V, Cr, As in the B-site whose bond distortions do not appear to be closely associated unlike the Ca and Cd apatites, which are closely associated. The lattice distortions appear to be a strong function of electronic structure interaction between Pb and B-site elements in Cl-apatites. In the apatite literature, this pattern is not known and has been unravelled for the first time using data mining in this work.

Topological observations made on the data were: Nature of the loadings on the principal components did not change much with a change in the p value of the distance metric. Since low-dimensional points obtained are different for both Isomap and PCA, it can be said that the apatite data lies on a nonlinear manifold in the embedding space.

2.6 Conclusion

In this paper, we have detailed a mathematical framework of selected nonlinear dimensionality reduction techniques for constructing reduced order models of complicated datasets and discussed

key questions involved in data selection. During that process we have also introduced the basic principles behind data dimensionality reduction and illustrated their use with the help of example apatite dataset in materials science using both linear and non-linear methods [3]. Another significant contribution of this paper is that we also describe a rigorous technique (based on graph-theoretic analysis) to estimate the optimal dimensionality of the low-dimensional (or parametric) representation. The techniques are packaged into a modular, computational scalable software framework with a graphical user interface - Scalable Extensible Toolkit for Dimensionality Reduction (SETDiR). This interface helps to separate out the mathematics and computational aspects from the scientific applications, thus significantly enhancing utility of DR techniques to the scientific community.

The applicability of this framework in constructing reduced order models of complicated materials dataset is illustrated with an example dataset of apatites described in structural descriptor space. Apatites($A_4^I A_6^{II}(BO_4)_6X_2$) have the ability to accommodate numerous chemical substitutions and hence can be used in the process of detoxification. SETDiR was applied to a dataset of 25 apatites being described by 29 of its structural descriptors. The corresponding low-dimensional plots revealed insights into the correlation between structural descriptors like ionic radius, covalence, etc; with properties like apatite stability. The plots also concluded that the shape of the surface on which the data lies is nonlinear. This information is crucial as it can promote the use of apatites as an antidote in lead poisoning.

2.7 Acknowledgements

This research was supported in part by the National Science Foundation through XSEDE resources provided by TACC under grant number TG-CTS110007, and supported in part by NSF PHY-0941576, and NSF-0917202.

[3] A comprehensive catalogue of nonlinear dimensionality reduction techniques along with the mathematical prerequisites for understanding dimensionality reduction could be found at: [84]

CHAPTER 3. PARALLEL FRAMEWORK FOR DIMENSIONALITY REDUCTION OF LARGE-SCALE DATASETS

A paper submitted to The Journal of Scientific Programming 2013

S. Samudrala, J. Zola, S.Aluru, B. Ganapathysubramanian

As a first author of this paper, I (S. Samudrala) developed the parallel dimensionality reduction framework, packaged into a software called **Pa**rallel **D**imensionality **Re**duction PaDRe with help from J.Zola under the supervision of B. Ganapathysubramanian and S. Aluru.

3.1 Abstract

Dimensionality reduction refers to a set of mathematical techniques used to digest the original high-dimensional data, while preserving its selected properties. Improvements in simulation strategies and experimental data collecting methods result in the deluge of heterogeneous and high-dimensional data, which often makes dimensionality reduction the only viable way to gain qualitative and quantitative understanding of the data. However, existing dimensionality reduction software do not scale to datasets arising in real-life applications that may consist of thousands of points lying in millions of dimensions. In this paper, we propose a parallel framework for dimensionality reduction of large-scale data. We identify key components underlying the spectral dimensionality reduction techniques, and we describe their efficient parallel implementation. We show that the resulting framework can be used to process datasets consisting of millions of points when executed on a 16,000-core cluster, which is beyond the reach of currently available tools. To

further demonstrate applicability of our framework we perform dimensionality reduction of 75,000 images representing morphology evolution during manufacturing of organic solar cells in order to identify the optimal processing parameters.

3.2 Introduction

Computational analysis of very high dimensional data continues to be a challenge and spurs the development of numerous techniques. An important and emerging class of techniques for dealing with such high dimensionality is dimensionality reduction. In many applications, features of interest can be preserved while mapping the high dimensionality data to a small number of dimensions and while preserving certain properties. Such mappings include popular techniques such as Principle Component Analysis (PCA) [88] and complex non-linear maps such as Isomap [136] and Kernel PCA [77].

Linear manifold learning techniques like PCA or Multi-dimensional scaling [137, 130, 80, 31] existed as an orthogonalization technique for several decades. Nonlinear methods like: Isomap, Locally Linear Embedding (LLE) [115], Hessian LLE (hLLE) [39], were discovered recently. Another league of methods that emerged in the past few years is the unsupervised learning techniques including artificial neural networks like Sammon's nonlinear map [118], Kohenen's or Self Organizing Maps (SOM) [76], Curvilinear Component Analysis [36], etc; Modifications to the existing algorithms of manifold learning, either to improve the efficiency or performance, was another area where efforts were focused [154, 33, 159, 135, 117]. For example, Landmark Isomap [131] is a modification to the original Isomap method to extend its usage to larger datasets by picking a few representative points and applying Isomap technique to it. Along with the emergence of new manifold learning techniques, there emerged simultaneously, different sequential implementations of these techniques on various platforms and in various programming languages [139, 140].

Dimensionality reduction techniques are often compute-intensive and do not easily scale to large datasets. Recent advances in high-throughput measurements using physical entities such as sensors or results of complex numerical simulations are generating data of extremely high dimensionality.

It is becoming increasingly difficult to process such data serially. In this paper, we propose a parallel framework for dimensionality reduction. Rather than focus on a particular dimensionality reduction method, we consider the class of spectral data decomposition methods. We perform a systematic analysis of these dimensionality reduction techniques and provide a unified view that can be exploited by dimensionality reduction algorithm designers. We identify common computational building blocks required for implementing spectral dimensionality reduction methods, and use these abstractions to derive a common parallel framework. Till date, little efforts have been made in developing parallel implementations of these dimensionality reduction methods; other than a version of PCA [161, 3] and algorithm development for GPU platforms [69, 155].

We design and implement such a parallel framework for dimensionality reduction that can handle large datasets, and which scales to thousands of processors. We demonstrate advantages of our software by analyzing 75,000 images of morphology evolution during manufacturing of organic solar cells, which enables us to identify the optimal fabrication parameters.

The remainder of this paper is organized as follows. In Section 3.3 we introduce the dimensionality reduction problem and describe basic spectral dimensionality reduction techniques, highlighting their computational kernels. In Section 3.4 we provide detailed description of our parallel framework including algorithmic solutions. Finally, in Section 3.5 we present experimental results and we conclude the paper in Section 3.6.

3.3 Materials and Methods

The problem of dimensionality reduction can be formulated as follows. Consider a set $X = \{x_0, x_1, \ldots, x_{n-1}\}$ of n points, where $x_i \in \mathbb{R}^D$, and $D \gg 1$. We are interested in finding a set $Y = \{y_0, y_1, \ldots, y_{n-1}\}$, such that $y_i \in \mathbb{R}^d$, $d \ll D$ and $\forall_{i,j}\ |x_i - x_j|_h = |y_i - y_j|_h$. Here, $|a-b|_h$ denotes a specific norm that captures properties we want to preserve during dimensionality reduction [84]. For instance, by defining h as Euclidean norm we preserve Euclidean distance, thus obtaining a reduction equivalent to the standard technique of Principal Component Analysis (PCA) [88]. Similarly, defining h to be the angular distance (or conformal distance [14]) results in Locally Linear

Embedding (LLE) [115] that preserves local angles between points. In a typical application [45, 152], x_i represents a state of the analyzed system, e.g. temperature field, concentration distribution, etc. Such state description can be derived from experimental sensor data or can be the result of a numerical simulation. However, irrespective of the source, it is characterized by high dimensionality, that is, D is typically of the order of 10^6 [151]. While x_i represents just a single state of the system, common data acquisition setups deliver large collections of such observations, which correspond to the temporal or parametric evolution of the system [45]. Thus, the cardinality n of the resulting set X is usually large ($n \sim 10^4$–10^5). Intuitively, information obfuscation increases with the data dimensionality. Therefore, in the process of Dimensionality Reduction (DR) we seek as small dimension d as possible, given constraints induced by the norm $|a - b|_h$ [84]. Routinely, $d < 4$ as it permits, for instance, visualization of the set Y.

DR techniques have been extensively researched over the last decade [84]. In particular, methods based on the spectral data decomposition have been very successful [88, 136, 39], and have been widely adopted. Early approaches in this category exploited simple linear structure of the data, e.g. PCA or Multidimensional Scaling (MDS) [81]. More recently techniques that can unravel complex non-linear structures in the data, for example Isomap [136], LLE, Kernel PCA [77], etc. have been developed. While all these methods have been proposed taking into account specific applications [140, 84], their underlying formulations share similar algorithmic mechanisms. In what follows we provide a more detailed overview of spectral DR techniques, and we identify their common computational kernels that form the basis for our parallel framework.

3.3.1 Spectral Dimensionality Reduction

The goal of DR is to identify low-dimensional representation Y of the original dataset X, that preserves certain predefined properties. The key idea underpinning spectral DR can be explained as follows. We encode desired information about X, i.e. topology or distance, in its entirety by considering all pairs of points in X. This encoding is represented as a matrix $A_{n\times n}$. Next, we subject matrix A to unitary transformation V, i.e. transformation that preserves norm of A, to obtain its sparsest form Λ, where $A = V\Lambda V^T$. Here, $\Lambda_{n\times n}$ is a diagonal matrix with rapidly

diminishing entries. As a result, it is sufficient to consider only d entries of Λ to capture all the information encoded in A. These d entries constitute the set Y. The above procedure hinges on the fact that unitary transformations preserve original properties of A [51]. Note also, that it requires a method to construct matrix A in the first place. Indeed, what differentiates various spectral methods is the way information is encoded in A.

We summarize the general idea of spectral DR in Algorithm 1. In the first four steps we construct the matrix A. As indicated, this matrix encodes information about the property that we wish to preserve in the process of DR. To obtain A we first identify the k nearest neighbors (KNN) of each point $x_i \in X$. This enables us to define a weighted graph G that encapsulates, both distance and topological, properties of the set X. Given graph G, we can construct a function $F_G : X \times X \to \mathbb{R}$ to isolate the desired property. For instance, consider the Isomap algorithm in which the geodesic distance is maintained. In this case, F_G returns the length of the shortest path between x_i and x_j in G. Note that for some methods F_G is very simple, e.g. for PCA it is equivalent to ω, $F_G(x_i, x_j) = \omega(x_i, x_j)$, while for other methods F_G can be more involved. Differences between various DR methods and their corresponding function F_G are outlined in Table 3.1. The property extracted by function F_G is stored in an auxiliary matrix W, which is next normalized to obtain matrix A. This process of normalization is a simple algebraic transformation, which ensures that A is centered, and hence, that the final low-dimensional set of points [Y] contains the origin and is not an affine translation [51]. Subsequently, A is spectrally decomposed into its eigenvalues that constitute the sparsest representation of A. Resulting eigenvectors and eigenvalues are then post-processed to extract a set Y of low-dimensional points.

The abstract representation of spectral DR methods in Algorithm 1 is based on a thorough analysis of existing techniques [88, 136, 115]. While this representation is compact, it offers sufficient flexibility to, for instance, design new dimensionality reduction procedures. At the same time it provides clear separation of individual computational steps, and explicates data flow in any DR process. We exploit both these facts when designing our parallel framework.

Input: Set $X = \{x_0, x_1, \ldots, x_{n-1}\}, x_i \in \mathbb{R}^D$, and the target dimension d.
Output: Set $Y = \{y_0, y_1, \ldots, y_{n-1}\}, y_i \in \mathbb{R}^d$.

1: For each $x_i \in X$ find its k nearest neighbors.
2: Define directed weighted graph $G = (X, E, \omega)$,
where $(x_i, x_j) \in E$ iff x_j is a neighbor of x_i,
and $\omega(x_i, x_j)$ is a distance measure,
usually $\omega(x_i, x_j) = |x_i - x_j|_2$.
3: Let $W_{ij} = F_G(x_i, x_j)$, where F_G extracts specific property
from graph G.
4: Normalize W to obtain matrix A.
5: Find eigenvectors of A, $A = V\Lambda V^T$.
6: Identify latent dimensionality d.
7: Y is represented by the first d rows of V.

Algorithm 1: Spectral Dimensionality Reduction.

Table 3.1 Comparison of selected spectral dimensionality reduction methods.

	PCA	Isomap	LLE
Parameter k in KNN	n	$\sim d$	$\sim d$
Function F_G	$\omega(x_i, x_j)$	Length of the shortest path between x_i and x_j in G.	α_{ij} if $(x_i, x_j) \in E$, 0 otherwise, where $x_i = \sum_{x_l:(x_i,x_l)\in E} \alpha_{il} x_l$.
Normalization	$W^*_{ij} = W^2_{ij}$, $A = H^T W^* H$	$W^*_{ij} = W^2_{ij}$, $A = H^T W^* H$	$A = (I - W)^{-1}(I - W)^{-T}$

Note: I is the identity matrix, and $H = I - \frac{1}{n}\mathbf{1}_{n\times n}$.

Table 3.2 Run time (in seconds) of different DR components.

n	100	1000	2000	4000
KNN	0.08640	1.34998	5.66768	27.91930
W in PCA	–	–	–	–
W in Isomap	0.06470	14.9030	130.130	1153.30
W in LLE	0.08960	0.12601	0.24609	0.49253
Normalize	0.00195	0.11875	0.74934	5.56630
Eigensolve	0.02916	0.05536	0.23267	0.85211
Extract Y	0.00020	0.00014	0.00016	0.00022

3.3.2 Performance Analysis of Dimensionality Reduction Methods

We used the above presentation of DR methods to identify their basic computational kernels. To better understand how these kernels contribute to the overall performance of different DR methods we performed a set of experiments using domain specific implementation in Matlab. Experiments were carried out for varying n and a fixed $D = 1000$ on a workstation with 8 GB of RAM and an Intel 3.2 GHz processor. Obtained results are presented in Table 3.2.

As can be seen, the run time of analyzed methods is dominated by two steps, namely KNN and construction of the auxiliary matrix W. Together they account for 99.8% of the total execution time for n=4000. In our implementation the KNN procedure depends on all-vs.-all distance calculations. This is justified taking into account that D is very large, and thus efficient algorithmic strategies for KNN, e.g. based on hierarchical space decomposition [56], are infeasible. Consequently, complexity of this step is $O(Dn^2)$. The cost of computing matrix W depends explicitly on the definition of function F_G. Among existing DR techniques this function is the most complex for the Isomap method. Recall, that in the process of DR we are interested in preserving either distance or local topology characteristics. Local topology properties can be directly obtained from KNN [115, 13, 39], inducing computationally efficient definition of F_G. Conversely, distance characteristics must conform to global constraints and therefore have higher computational complexity [123]. In case of Isomap, pairwise geodesic distances can be efficiently derived from all-pairs shortest path distances using e.g. Floyd-Warshall algorithm, with $O(n^3)$ worst-case complexity.

Another significant DR component is normalization. Although implementation of this step varies between different methods it is invariably dominated by matrix-matrix multiplication. Therefore, we assume overall normalization complexity to be $O(n^3)$. The last important component is the eigenvalue solver. In general complexity of this kernel varies depending on the particular solver used. Commonly employed algorithms include Lanczos method [82], Krylov sub-space methods [116], or deflation based power methods [148, 104]. The choice of method is driven by the structure of the matrix and the number of required eigenvalues. Standard distance preserving DR methods operate on dense symmetric matrices, while topology preserving methods involve sparse symmetric

matrices. Accordingly, complexity of these techniques is usually $O(dn^2)$, where d is the number of desired eigenvalues.

A final key factor we have to consider is memory complexity of the described kernels. Here, the main contributing structure are matrices W and A. These matrices are most often dense, and in the majority of cases require $O(n^2)$ storage. Because KNN directly depends on distances between all pairs it utilizes a $n \times n$ matrix as well. Finally, input dataset X requires $O(Dn)$ memory.

One important caveat that affects the above analysis is the relationship between D and n. In many applications D is significantly greater than n. This is not surprising taking into account that acquiring high resolution data (hence high dimensional) is resource intensive. Therefore one may expect that with increasing D there is rapid decrease of n. In our applications [151, 55] it is not uncommon that $D = O(n^2)$ or even $D = O(n^4)$. Consequently, the KNN step in Algorithm 1 becomes the most compute intensive while memory requirement is dominated by the input data. Observe that this trend is reflected in our experimental data.

3.4 Parallel Framework for Dimensionality Reduction

Dimensionality reduction very quickly becomes both memory and compute prohibitive, irrespective of the particular method. Memory consumption arises from the size of input data and the auxiliary matrices created in the process. The computational cost is dominated by pairwise computations and weight matrix construction. The goal of our framework is to scale DR methods to very large datasets that could be analyzed on large parallel machines.

We designed our parallel DR package following the general outline presented in Algorithm 1. Taking into account significant memory and computational complexity we focused on distributed memory machines with MPI. To ensure modularity of the framework without sacrificing performance and scalability, we decided to employ a scheme in which processors are organized into a logical 2D mesh. In what follows, we assume a simple point-to-point communication model with latency τ and bandwidth $\frac{1}{\mu}$.

3.4.1 Constructing Graph G

The graph construction procedure is based on identifying k nearest neighbors of each input point. Because of the high dimensionality of the input data, it is advantageous to implement KNN in two steps where we first compute all pairwise distances and then we identify neighbors in a simple scan. Note that these pairwise distances actually represent the distance measure ω (see Algorithm 1). Therefore, we will consider ω to be a $n \times n$ distance matrix. Parallel pairwise computation is a well studied problem [60]. Here, we benefit from our earlier experience with accelerating pairwise computations on heterogeneous parallel processors [122].

Let $p = q^2$ denote the number of processors conceptualized as organized into a $q \times q$ virtual mesh. We decompose ω into blocks of $\frac{n}{q} \times \frac{n}{q}$ elements. Processor with coordinates (i, j) is responsible for computing elements of ω within block (i, j). This scheme requires that each processor store two blocks of $\frac{n}{q}$ points of the input dataset X, that correspond to row-vectors and column-vectors used to compute respective part of the matrix ω. In our implementation, the distribution of the input dataset is performed by parallel I/O with initially preprocessed X. Note that to obtain a single element of ω we perform $|a - b|_2$ norm computations, which are particularly well suited for vectorization. Therefore, we hand-tuned our code to benefit from the SSE extension of modern processors.

Given pairwise distances, the second step is to identify neighbors of individual points (i.e., vertices of G). This step is executed only for methods where $k < n$, which virtually involves all methods other than PCA (see Table 3.1). As in the case of pairwise computations, it can be efficiently parallelized using the following scheme. Initially, each processor creates a set of k candidate neighbors with respect to the block of matrix ω it stores. Specifically, processor with coordinates (i, j) searches for neighbors of the set of points $\left[x_{\frac{in}{q}}, \ldots, x_{\frac{(i+1)n}{q}}\right)$ by analyzing rows of its local block of ω. Because k is very small this operation can be performed using a simple linear scan. Next, all processors within the same row perform all-to-all communication to aggregate candidate neighbors. Here, the candidate neighborhood of point x_l is assembled on the processor with coordinates $\left(\left\lfloor \frac{lq}{n} \right\rfloor, \left\lfloor \frac{lp}{n} \right\rfloor \mod q\right)$. This processor merges candidate neighborhood lists into the

final set of k nearest neighbors. Observe that this completes the graph construction phase - graph G is now stored in the form of adjacency list distributed over p processors. The computational complexity of the entire procedure is $O\left(\frac{Dn^2}{p} + (\tau + \mu\frac{nk}{p})\sqrt{p}\right)$, which is optimal for $p < n^2$.

3.4.2 Building Auxiliary Matrix W

Given graph G we proceed to the next step, which involves constructing the auxiliary matrix W from the information encapsulated in G. As is the case of ω we distribute W over $q \times q$ mesh of processors.

Recall that the formulation of DR methods proposed in Algorithm 1 ensures that the only step that is method dependent is the construction of matrix W. Consequently, any parallel implementation of this step will vary, but it will reflect limitations inherent to the sequential counterpart. Specifically, topology preserving methods, such as e.g. LLE, will involve only local data, and hence will be amenable to embarrassing parallelism with limited or no communication. Conversely, distance preserving methods will inevitably require a global data view, and thus potentially more sophisticated parallelization strategies. Following our previous claim regarding complexity of Isomap we focus our presentation on the parallel implementation of this particular method.

The function F_G used in Isomap is based on the geodesic distance, which has been mathematically shown to be asymptotically equivalent to graph distance in G (i.e., shortest path distance) [15]. However, the geodesic distance is a metric, while all-pairs shortest path in directed graph G does not have to satisfy the symmetry condition. Therefore, to obtain W, special attention must be paid to how shortest path distances are used. More precisely, graph G must be transformed to ensure that it is symmetric. Note that after such transformation the graph is no longer regular, i.e., certain nodes may have more than k neighbors (see Figure 3.1).

Taking into account the above requirements we obtain the following procedure of constructing W in parallel. First, all processors within the same row perform all-to-all communication to replicate graph G. As a result, each column of processors stores a copy of the entire graph G that is row-wise distributed between q processors in that column. Thanks to this, each processor can initialize its local part of W without further communication. After initialization W represents the distributed

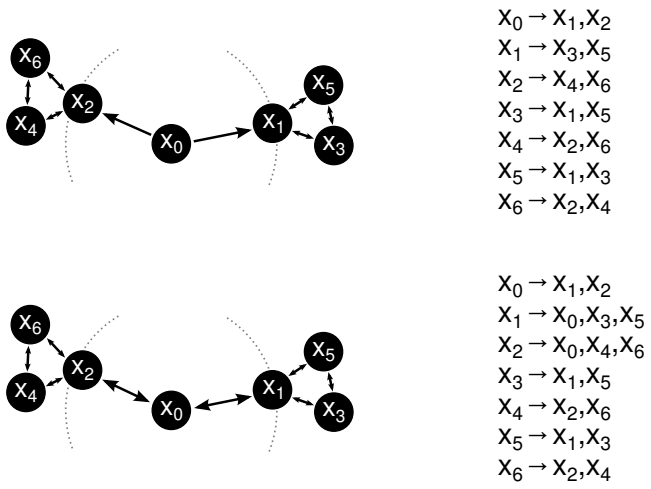

Figure 3.1 Graph G before (top) and after (bottom) symmetrization for an example set of 7 points and $k = 2$.

adjacency matrix of G, where $W_{ij} = \omega(x_i, x_j)$ if x_j is a neighbor of x_i, and $+\infty$ otherwise. In the next step symmetrization procedure is executed. Processors with coordinates (i, j) and (j, i), where $i \neq j$, exchange respective blocks of W, and select element-wise minimum value between blocks. Similar operation is performed locally by processors on the diagonal, i.e., processors for which $i = j$. At this stage W can be used to identify all-pairs shortest paths. Several parallel algorithms have been proposed to address this problem, including on the PRAM model [28, 73, 95], shared memory architectures [72], multi/many cores [1, 35, 75] as well as distributed memory machines [67, 95]. Amongst the existing parallel strategies we decided to adopt the checkered-board version of the parallel Floyd's algorithm [72]. Briefly, the method proceeds in n iterations, where in each iteration every processor performs $O\left(\frac{n^2}{p}\right)$ operations to update its local block of W. All processors are synchronized at the end of each iteration, owing to the fact that in iteration l each processor requires l-th row and l-th column of W. After n iterations matrix W stores all-pairs shortest path, which concludes the entire procedure.

Complexity of this phase is dominated by the parallel Floyd's algorithm. While replication and symmetrization of G can be executed efficiently in $O\left(\frac{n^2}{p} + \tau + \mu\frac{n^2}{p}\right)$ time, all-pairs path searching involves extensive communication that slightly hinders scalability. Nevertheless, the algorithm remains scalable as long as $p < \frac{n}{log(n)}$, with overall complexity $O\left(\frac{n^3}{p} + n\left(\tau + \mu n \log(p)\right)\right)$.

3.4.3 Matrix Normalization

The goal of normalization is to transform matrix W such that resulting matrix A is both row and column centered, i.e., $\sum_i A_{ij} = 0$ and $\sum_j A_{ij} = 0$. The normalization stage in all cases consists of matrix-matrix multiplication (see Table 3.1). However, in certain situations, especially in distance-preserving methods, explicit matrix multiplication can be avoided by taking advantage of structural properties of one of the matrices (e.g. the matrix H in Table 3.1). This is the case for, e.g. PCA and Isomap, where we exploit the fact that matrices H and H^T are given analytically, and thus can be generated in-place on each processor that requires them to perform multiplication. Consequently, the communication pattern inherent to the standard parallel matrix-matrix multiplication algorithms is simplified to one parallel reduction in the final dot-product operation. The complexity of this approach is $O\left(\frac{n^3}{p} + (\tau + \mu\frac{n^2}{p})\sqrt{p}\right)$.

3.4.4 Finding Eigenvalues

Computing eigenvalues is the final step in the dimensionality reduction process. Although, parallel eigensolvers are readily available, they are usually designed for shared-memory and multi/many-core architectures [25, 10, 18, 37]. This unfortunately makes them impractical for our purposes. At the same time, existing distributed memory solutions are not scalable and cannot handle large and dense data. For instance, one of the more popular packages, SLEPc [61], uses a simple 1D decomposition and in our tests did not scale to more than 4096 processors. A more recent library, elemental [70], which is still under development, offers 2D block-cyclic decomposition, but relies on a fixed block size (private communication). For these reasons we decided to implement a custom eigenvalue solver that exploits special properties of matrix A (symmetric, positive semi-definite), and computes only the first d eigenvalues. Our solver is based on the power method [51] and matrix deflation and is outlined in Algorithm 2. Note that power methods are considered easy, but not efficient to parallelize. At the same time, however, they are at heart of several important real-life systems, for instance, Google's PageRank [2].

In general, our approach follows the standard scheme of power method (lines 3–18), repeated d

Input: Matrix $A_{n\times n}$ and the required numbed of eigenvalues d.
2D mesh of $p = q \times q$ processors.
Output: Set of eigenvalues and eigenvectors of A,
$\Lambda_{0..d-1} = \{\lambda_0, \lambda_1, \ldots, \lambda_{d-1}\}$ and $V_{0..d-1} = \{v_0, v_1, \ldots, v_{d-1}\}$.

1: Let $\mathbf{x}$ be a column-wise distributed vector in $\mathbb{R}^n$.
2: **for** $i \leftarrow 1 : d$ **do**
3: Initialize $\mathbf{x}$ randomly. Processors within the same column use the same seed.
4: $column \leftarrow true$
5: **while** *not converged* **do**
6: Compute $z = A\mathbf{x}$ locally.
7: **if** $column = true$ **then**
8: Perform column-wise all-reduce to obtain z.
9: **else**
10: Perform row-wise all-reduce to obtain z.
11: **end if**
12: $column \leftarrow \neg column$
13: $\mathbf{x} \leftarrow z$
14: **end while**
15: Compute $u = Az$ as in steps 6–11.
16: Replicate entire vector u and z on each processor.
17: $\lambda_i \leftarrow \frac{z \cdot u}{z \cdot z}$
18: $v_i \leftarrow \frac{z}{|z|_2}$
19: Deflate local block of A: $A \leftarrow A - \lambda_i v_i v_i^T$.
20: **end for**

Algorithm 2: 2D-Block Parallel Power Method.

times to identify first d largest eigenvalues. After identification of an eigenvalue and its associated eigenvector, the matrix A is deflated – the contribution of the vector is removed from A (line 19). Observe that power method involves nested matrix-vector product (lines 6–11) that under normal circumstances would require parallel vector transposition. However, our parallel implementation benefits from the fact that A is symmetric, hence eliminating need for vector transposition. Indeed, the entire procedure consists of local matrix-vector product followed by all-reduce operation. Here, the reduction operation alternates between columns and rows as needed to ensure that vector $\mathbf{x}$ is stored properly. Note that the power method is bounded by convergence criteria (line 5). In our case we use one of several popular conditions, which involves checking relative error between the current and previous estimate of the eigenvalue that can be performed every several iterations. We also note that convergence is significantly improved by using a matrix shifting strategy in the form $A = A - \delta I$, where δ is a positive number [92].

Extracting eigenvalue and eigenvector in iteration i (lines 11–18) depends on vectors u and z, while deflation step involves λ_i. Therefore, it is advantageous to replicate both u and z in their entirety on each processor. We achieve this with all-to-all communication executed by processors within the same row. This allows us to execute the deflation step in parallel, with each processor updating its local block of matrix A. Thus, the complexity of a single iteration of the power method is $O\left(\frac{n^3}{p} + \left(\tau + \mu\frac{n}{\sqrt{p}}\right)\log(p)\right)$, while the deflation step is $O\left(\frac{n^2}{p} + \tau\sqrt{p} + \mu n\right)$.

To conclude this section we would like to emphasize that our solver operates under the same assumptions as any power method. It requires that the first d eigenvectors of A are linearly independent, the initial vector $\mathbf{x}$ generated in i-th iteration is not orthogonal to the eigenvector v_i, and finally, the first d eigenvalues are non-degenerate [51]. Note that these conditions are not restrictive and are easily satisfied in the context of dimensionality reduction.

3.5 Results and Discussion

To assess scalability of our framework and test its performance in real-life applications, we performed a set of experiments using the Ranger cluster [30]. A single node of this machine is based

Table 3.3 Run time in seconds for different p, and varying problem size n. Due to memory limitations problem with $n = 32768$ cannot be solved on less than $p = 256$ processors.

p	n			
	4096	8192	16384	32768
16	404.63	3492.28	45288.93	–
64	101.72	761.75	6906.64	–
256	33.99	263.24	1655.39	14613.33
1024	39.06	124.19	682.91	3964.65

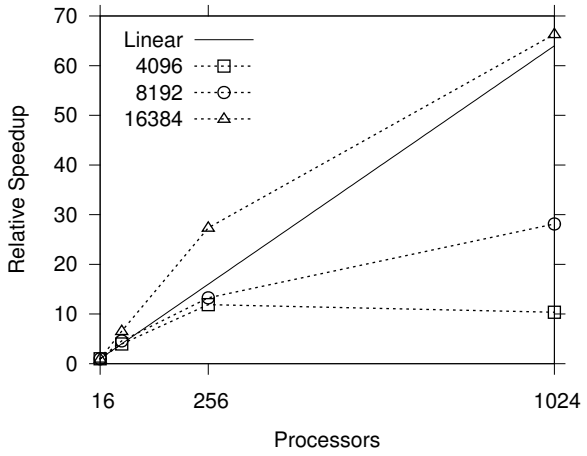

Figure 3.2 Relative speedup for different problem sizes.

on AMD processors working at 2.3 GHz, and provides 16 cores with 32 GB of DDR2 RAM, and 512 KB of L2 cache per core. Nodes are connected by a multi-stage Infiniband network that offers 1 Gbit/s bandwidth. To compile all test programs and the framework we used the Intel C++ 10.1 compiler with standard optimization flags, and MVAPICH 1.0.1 MPI implementation. In every test we ran one MPI process per CPU core, which we refer to as processor.

3.5.1 Scalability Tests

In the first set of experiments we measured how problem size influences performance of our solution. We created a collection of synthetic datasets consisting of $n = \{4096, 8192, 16384, 32768\}$ points with $D = 10000$. Next, we performed Isomap dimensionality reduction using different number of processors. Obtained results are summarized in Table 3.3 and Figure 3.2.

Table 3.4 Component-wise run time in seconds for varying problem size and $p = 1024$ and $D = 10000$.

n	2048	4096	8192	16384	32768
KNN	0.623	1.389	5.721	22.254	86.706
W in Isomap	9.132	56.517	128.225	457.306	1697.124
Normalize	0.160	0.905	6.526	223.240	2546.11
Eigensolve	0.050	0.155	0.188	0.699	2.838

The results show that our framework provides very good scalability for large problem sizes irrespective of the number of processors used. The super-linear speedup observed for $n = 16384$ is naturally explained by cache performance. Observe that the dominating computational factors in our framework are operations like matrix-matrix and matrix-vector products, which are well suited to exploit memory hierarchy. A slightly weaker performance for small problem sizes and large number of processors can be attributed to network latency that offsets computational gains.

To further understand how different components of the framework perform, we measured their run time obtained for changing problem sizes. Table 3.4 shows that all modules scale as we would expect based on their theoretical complexity. The most time consuming stages are construction of the auxiliary matrix W for Isomap and normalization. This is not surprising taking into account that both components scale as $O(n^3)$, and the parallel Floyd's algorithm involves n rounds of communication. The abrupt performance decrease in the normalization stage, which can be observed for $n = 16384$, can be attributed to cache performance. Recall that normalization depends on matrix-matrix multiplication, and hence is inherently sensitive to data locality. The final remark concerns k nearest neighbors module and eigenvalue solver. The KNN scales linearly with the data dimension D (see Table 3.5), and both modules can be used as standalone replacements whenever KNN or d largest eigenvalues problem has to be solved.

In the final test we compared our parallel eigensolver with SLEPc [61], one of the most popular and widely used libraries providing eigensolvers. SLEPc is an efficient and portable framework that offers intuitive user interface. In many cases it is the first choice for solving large scale sparse eigenvalue problems.

Table 3.5 Run time in seconds of KNN component for $n = 1024$ and different number of processors and varying D.

p	D			
	100	1000	10000	100000
16	0.053	0.530	5.466	92.014
64	0.015	0.115	1.373	22.984
256	0.005	0.027	0.349	5.860
1024	0.002	0.007	0.682	1.875

Table 3.6 Comparison of PaDRe and SLEPc. For $p = 1024$ SLEPc failed to execute.

p	$n = 1024$		$n = 4096$	
	PaDRe	SLEPc	PaDRe	SLEPc
16	0.0444	4.8159	2.5315	0.7049
64	0.0088	2.1666	0.6056	0.8134
256	0.0705	8.5538	0.1251	2.4143
1024	0.0742	*	0.1320	*
4096	0.0411	N/A	0.2024	10.9992

Table 3.6 shows that our implementation systematically outperforms SLEPc. This can be explained by two main factors: first, unlike SLEPc our implementation follows 2D data decomposition scheme, which offers better scalability, and second, we are seeking only the d largest eigenvalues.

3.5.2 Using dimensionality reduction to explore manufacturing pathways

Solar cells (or plastic solar cells) manufactured from organic blends (i.e., a blend of two polymers) represent a promising low-cost, rapidly deployable strategy for harnessing solar energy. While highly cost-effective and flexible, their low power conversion efficiency makes them less competitive on a commercial scale in comparison with conventional inorganic solar cells. A key aspect determining the power conversion efficiency of organic solar cells is the morphological distribution of the two polymer regions in the device. Recent studies reveal that significant improvement in power conversion efficiency is possible through better morphology control of the organic thin film layer during the manufacturing process [151, 24, 65, 107, 124, 34, 103]. High-throughput exploration of the various manufacturing parameters (evaporation rate, blend ratio, substrate patterning

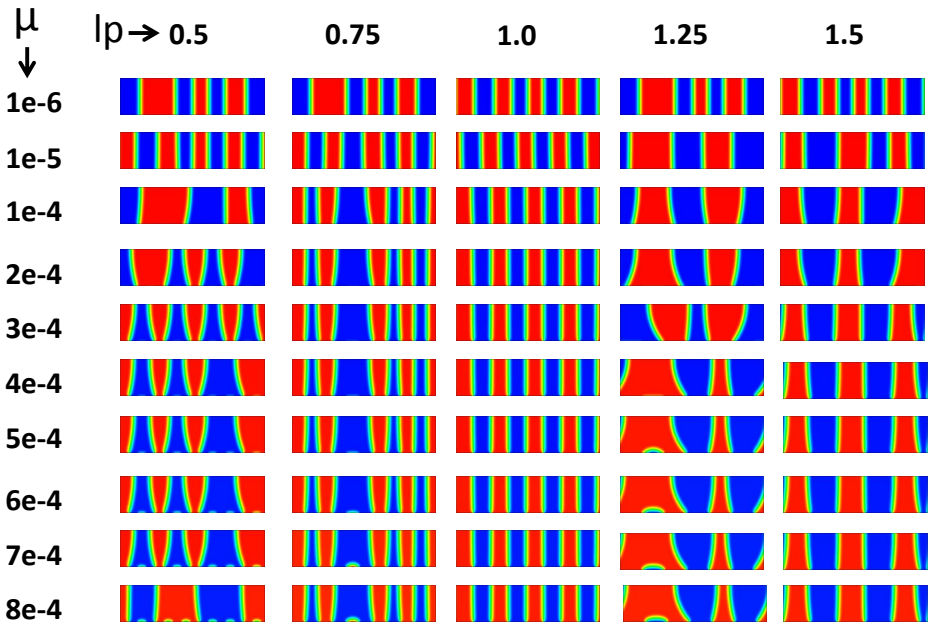

Figure 3.3 Snapshots of Microstructures representing final morphologies of 50 different processes under consideration

frequency, substrate patterning intensity, solvent) can potentially unravel process-morphology relationships that can help tailor processing pathways to obtain enhanced morphologies. Note that such high-throughput analysis results in data-sets that are too large to visually look for trends and relationships. A promising approach towards unraveling process-morphology relationships in this high-throughput data is via data-dimensionality reduction. We showcase the parallel framework on this pressing scientific problem. In particular, we focus on using dimensionality reduction to understand the effects of substrate patterning (patterning frequency and intensity) on morphology evolution. [1]

The dataset consists of $n = 75150$ morphologies. Each morphology is a 2-dimensional snapshot which is vectorized to have dimensionality $D = 8326$. Fig. 3.3 shows several representative final morphologies obtained by varying (a) the patterning frequency, lp, from 0.5 to 1.50, and (b) the intensity of the attraction/repulsion, μ, from $1 + 1e - 6$ to $1 + 8e - 4$. In all these cases, the

[1] Nano-tip photo-lithography patterning of the substrate has shown significant potential to guide morphology evolution [26]

lower surface of the domain is patterned to attract and/or repel specific classes of polymers, thus affecting the morphology. We performed dimensionality-reduction on this data set using $p = 16384$ processors on TACC Ranger. The total run time was 1058.4 seconds.

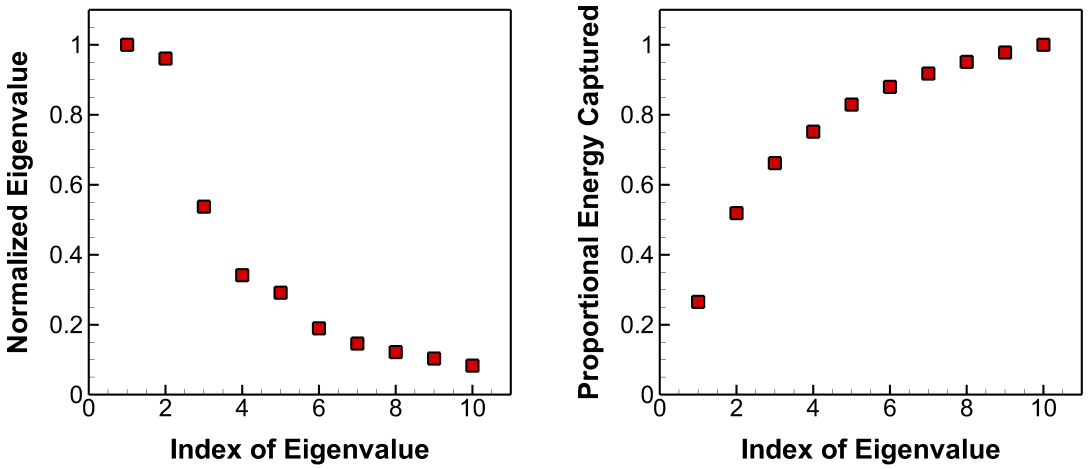

Figure 3.4 (A)Scree Plot for largest 10 Eigenvalues (B)Proportional Energy covered by first 10 Eigenvalues

Fig 3.4 plots the first 10 eigenvalues of the data. Note that the first three eigenvalues (and hence the first three principle components of the data) represent $\sim 70\%$ of the information content of the entire data. We therefore characterize each morphology in terms of this three dimensional representation.

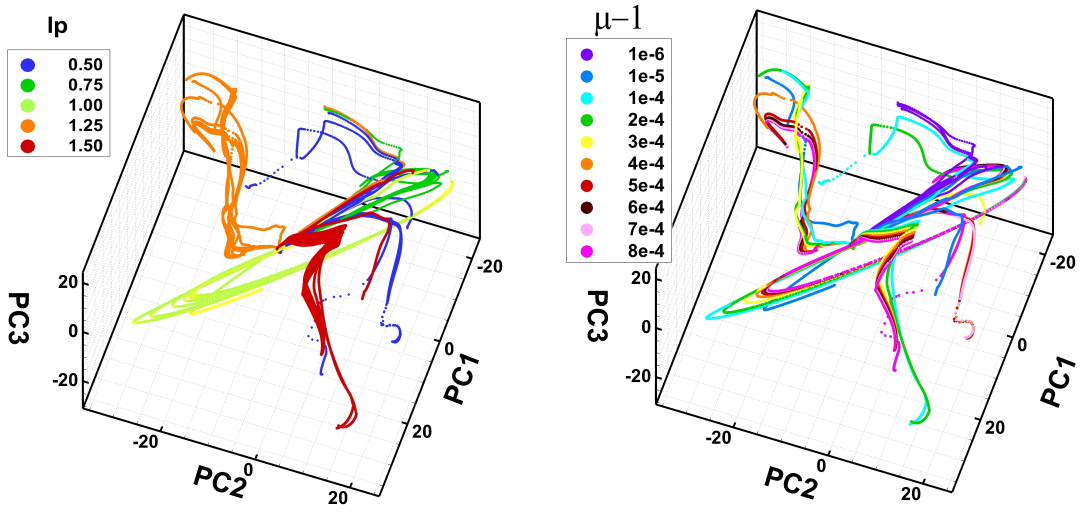

Figure 3.5 Morphology evolution with respect to the first 3 Prinicipal Components color coded with respect to (A) Patterning Frequency (lp), (B)Patterning Intensity (μ)

Fig 3.5 represents *all the morphologies* on this three dimensional reduced space. In Fig 3.5(A), the points are color coded according to the patterning frequency used, while in Fig 3.5(B), the points are color coded according to the patterning intensity used. This plot provides significant visual insight into the effects of patterning frequency and intensity.

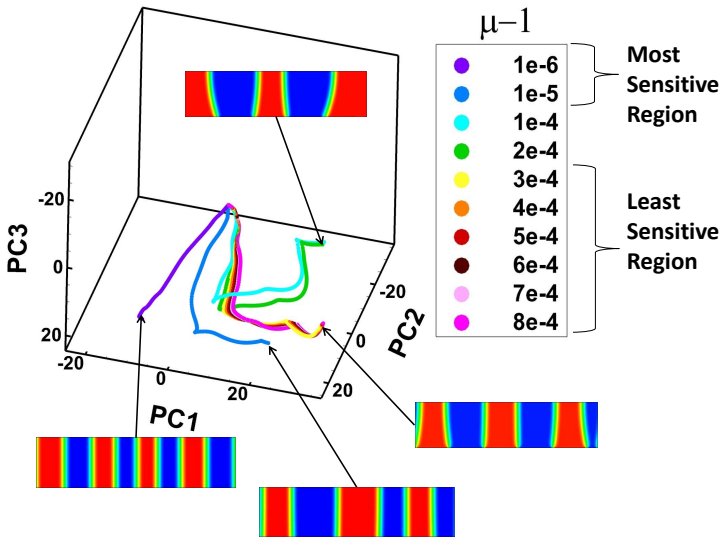

Figure 3.6 Morphology evolution in $lp = 1.50$: Categorization of parametric space

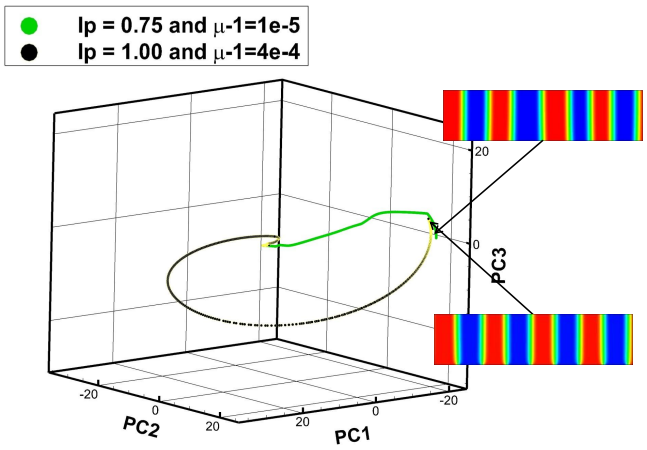

Figure 3.7 Multiple Pathways to Morphology Evolution

There exists a central plane of patterning frequency where the morphology evolution is highly regulated irrespective of the patterning intensity ($lp \leq 1$). This is particularly valuable information as the patterning frequency is much easier to control than patterning intensity from a manufacturing perspective. For patterning frequencies above $lp = 1$, the morphologies are highly sensitive to slight variations in both frequency and intensity. This is also clearly seen in Fig 3.6, where slight variations in the intensity give dramatically different final morphologies. Notice also the key insight that higher intensity do not necessarily give different morphologies. This allows us to preclude further (expensive) exploration of the phase space of increasing patterning intensity.

Finally, the low-dimensional plots also illustrate the ability to achieve the same morphology using different processing conditions. For instance, in Fig 3.7, we isolate the morphology evolution under two processing conditions that result in an identical morphology. Such correlations - most sensitive regions, least sensitive regions (Fig 3.6), configurations resulting in identical morphologies - are enormously useful as we tailor processing pathways to achieve designer morphologies. This analysis illustrates the power of parallel data-dimensionality reduction methods to achieve this goal. We defer a comprehensive physics based analysis of this data set to a subsequent publication.

3.6 Conclusion

In this work we illustrate a systematic analysis of dimensionality reduction techniques and recast them into a unified view that can be exploited by dimensionality reduction algorithm designers. We subsequently identified the common computational building blocks required to implement a spectral dimensionality reduction method. We used this insight to design and implement a parallel framework for dimensionality reduction that can handle large datasets, and scales to thousands of processors. We demonstrated the capability and scalability of this framework on several test datasets. We finally showcased the applicability and potential of the framework towards unraveling complex process-morphology relationships in the manufacture of plastic solar cells.

3.7 Acknowledgements

This research was supported in part by the National Science Foundation through XSEDE resources provided by TACC under grant number TG-CTS110007, and supported in part by NSF PHY-0941576, and NSF-0917202.

CHAPTER 4. A GRAPH-THEORETIC APPROACH FOR CHARACTERIZATION OF PRECIPITATES FROM ATOM PROBE TOMOGRAPHY DATA

A paper under-review with The Journal of Computational Materials Science

S. Samudrala, O. Wodo, S. K. Suram, S. Broderick, K. Rajan, B. Ganapathysubramanian

As a first author of this paper, I (S. Samudrala) developed a graph-based computational framework built upon another graph-based framework developed by O.Wodo under the supervision of B. Ganapathysubramanian. I also performed quantitative analysis of the results obtained by applying the current framework to a heterogenous atom probe data of Al-Mg-Sc prepared by experimentalists by S. K. Suram and S. Broderick under the supervision of K. Rajan.

4.1 Abstract

Atom Probe Tomography (APT) represents a revolutionary characterization tool that allows direct-space three-dimensional, atomic-scale resolution imaging along with the chemical identities of each detected atom. Quantitative analysis of APT data to perform characterization of precipitates in alloys gives clear insights into the structure-property relationships and helps in achieving the larger goal of materials-by-design. Most techniques currently used to extract precipitate topology and interface information from APT data are efficient; however, they are based on homogenization of the rich point cloud data which is inherently lossy. Furthermore, these methods require a specified, usually heuristic, concentration-level to draw iso-contours in order to extract characteristics of the

precipitate topology. These twin issues of homogenization and heuristics are compelling rationale for the development of a robust, scalable, heuristic-free, graph-based framework, which we call **Gra**ph methods for **P**recipitate **Top**ology Characterization (GraPTop). This framework is motivated by the equivalence between a 3D point cloud data of atoms and an undirected, weighted, and labeled graph. By considering the 3D point cloud data as an undirected, weighted, and labeled graph, we leverage powerful graph-based algorithms to identify the local topology of precipitates without the necessity of any heuristics. Since GraPTop is based on nearly linear-complexity graph-algorithms, it is scalable to extremely large data sets. Furthermore, the performance of this framework is insensitive to the complexity of the geometry or the number of the precipitates in the point cloud data. We showcase this framework by analyzing several regions of interest in a point cloud Al-Mg-Sc (Aluminium-Magnesium-Scandium) specimen APT data and extract several interesting measures describing the precipitate topology like area, volume, and nonconvexity.

4.2 Introduction

Atom Probe Tomography (APT) [98], [96] represents a revolutionary characterization tool for material scientists by providing direct-space three-dimensional, atomic-scale resolution with chemical identities of all the detected atoms. It involves controlled removal of atoms from a specimen's surface by field evaporation and then sequentially imaging and analyzing them with a TOF (Time of Flight) mass spectrometer. This technique currently provides the highest spatial resolution of any microanalysis technique. This capability provides a unique opportunity to experimentally study – with atomic resolution – chemical clustering and 3-D distributions of atoms; and directly test and refine atomic and molecular based modelling studies. While APT is a powerful technique with the capacity to gather information containing hundreds of millions of atoms from a single specimen, the ability to effectively use this information creates significant challenges. The main technological bottleneck lies in handling the extraordinarily large amounts of data in a reasonable amount of time [112]. This imposes a constraint for any quantitative technique to analyze the data to be both scalable and efficient.

One key material-science problem that can be analyzed using the APT is the characterization of precipitates in multi-component systems [112]. Of particular interest is the analysis and classification of precipitate topology, shape, size distributions as well as their interfacial properties [79, 125, 129, 128, 78, 127]. Addressing this problem can give clear insights into structure-property relationships (especially in the context of energy storage devices) and, thus, help in achieving a larger goal of accelerated materials-by-design [150]. Recent work in detecting nano-scale bio-geo-chemical interfaces [114] also show the increasing relevance and applicability of Atom Probe Tomography in fields outside materials science.

There exist several contemporary techniques for analyzing precipitates and clusters in 3D atom probe data. These chemical clusters are normally defined by calculating the concentration of different elements in the sample across a chosen bin size [59, 144], through the use of nearest neighbor approaches [132, 50], and cluster finding approaches [89, 142, 21]. Specifically, some of the approaches for precipitate analysis include: (i) proximity histograms, (ii) Fourier analysis, and (iii) friends-of-friends analysis. While chemical analysis of a chosen bin size through proximity histograms is a convenient technique for linking spatial features with chemistry and has shown good results [46, 110], the definition of the region analyzed is mathematically arbitrary. The region is defined by a user-defined chemical threshold value, and therefore a precipitate is defined largely through assumption. Additionally, the concentration is calculated as an averaging of the voxels comprising the region, thereby reducing the information resolution. Fourier analysis of APT data has previously been performed for analyzing regions of interest with the objective of identifying crystallographic structure [143, 145]. Among the outputs of this approach are mean precipitate size, shape and composition. While this approach is well suited for crystallographic analysis, the extensive computer memory requirements and limited resolution away from the poles prevents the ability to define the detailed shape and interface of precipitates [48]. Finally, a friends-of-friends approach for analyzing phases has shown promise [68, 97]. This approach is based on the finding solute atoms that are nearer to each other in the solute phase as opposed to the matrix. However, this approach is inefficient for high solute concentrations, and also has been found to falsely identify clusters due to bridging effects and insensitivities of parameters [133].

The standard approaches for defining precipitates, such as through proximity histograms or cluster analyses, result in precipitates defined with largely convex surfaces. For example, defining precipitate regions based on atomic clustering requires inputting a parameter defining the maximum distance allowed between atoms within the same cluster and an additional envelope parameter which effectively serves as defining the convex hull of the precipitate. However, even a small number of atoms within the cutoff distance can extend the envelope region to incorporate a region of limited solute concentration. By defining convex volumes, either regions of low solute concentration are included in the defined precipitate, or conversely high solute regions are omitted. The ability to define a non-convex surface for precipitates is necessary because the morphology of a cluster within this convex hull can be defined only by capturing the non-convex surfaces of these precipitates. Parameters traditionally measured from convex surfaces include volume and surface area, while non-convex parameters capture information including degree of kinetic coagulation. For example, clusters that are at initial stages of kinetic coagulation will have large non-convexity.

Convex precipitates define the minimum volume necessary to envelope the cluster. However, the information regarding the morphology of the cluster within this volume can be captured only by studying the non-convex nature of these precipitates. Convex parameters define traditionally studied parameters of clusters such as size, surface area. Whereas, non-convex parameters capture information such as degree of kinetic coagulation. For example, clusters that are at initial stages of kinetic coagulation will have large non-convexity. ROI 3 is an example of a cluster that is formed by initial stage coagulation of two clusters.

Most current approaches to performing precipitate analysis on APT point cloud data are either based on homogenization, or are dependent on heuristics to characterize the precipitates. These issues motivate the development of an efficient and heuristic-free method for performing the characterization of precipitates that can directly work with the point cloud (APT) data without homogenizing it. In this work, we detail a method of performing cluster selection and surface construction using a graph-based formalism that is heuristic-free, works directly with the point cloud data without homogenization into concentration fields, is very scalable to analyze very large data-sets, and is applicable to a wide range of chemistries, environments, and geometries. We call

this framework, **Gra**ph based methods for **P**recipitate **Top**ology Characterization (GraPTop). The following observations motivated our choice of a graph-based approach:

- The APT point-cloud data of atomic positions and their chemistry can be equivalently represented using an undirected, weighted, and labeled graph. Each atom becomes a graph vertex with a label denoting its chemistry. Each vertex is connected to its neighboring vertices through edges whose weight is proportional to the distance between the vertices (atoms). Fig. 4.1 shows a simple example of this concept.

- Most precipitate characterization properties (like size, shape, number of atoms, bounding shape, etc.) can be naturally recast as estimating properties of the equivalent graph.

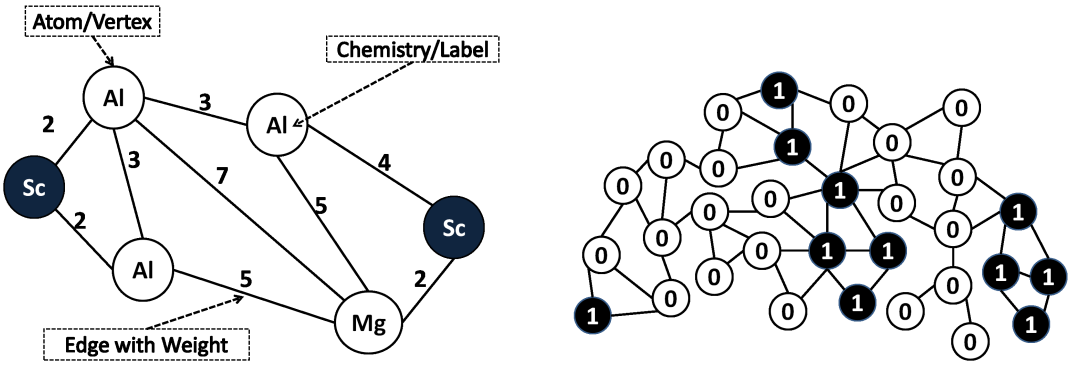

Figure 4.1 (a) Simple example of a Al-Mg-Sc alloy illustrating the equivalence between a graph and point cloud data (b) A larger example where the precipitate is labeled black and the solvent is labeled white.

Furthermore, a graph-based formalism is ideally suited for large-scale APT data sets, particularly due to the fact that:

1. Graph based methods are well-studied and have fast and efficient algorithms – for computing neighborhood and distance information – that are important for precipitate characterization. Furthermore, a graph approach directly works on the point cloud data without homogenizing it.

2. Graph based methods are easily scalable and hence, can be easily extended to larger problem sizes. Given the fact that APT deals with atomic scales, even for a moderately dense material

specimen a minute increase in the dimensions of the region of interest can cause an exponential increase in the size of the dataset. In such cases, scalability of the method becomes a critical factor in qualifying the applicability of a technique.

3. A graph-based approach is generic. That is, by making modifications to the definitions of parameters like edges, weights, and labels, different problems relating to the physical process can be solved. For example, while we focus on extracting precipitate shapes in this work, by replacing the Euclidean distance with radial distance (as weight definition) one can study the radial distribution of atoms in a precipitate. We have recently used such analysis to characterize the morphology of thin film organic photovoltaic [150, 149].

The outline of this paper is as follows: Section 4.3 gives an overview, methodology and associated algorithmic implementations of the framework. In section 4.4 we present results obtained by applying GraPTop to three different regions of interests in a point cloud data of an Al-Mg-Sc alloy. Quantified variables of the topology include area, volume and a measure of non-convexity of the scandium precipitates in the APT data. We conclude in section 4.5.

While working on this paper, we found a recent work based on Delaunay cluster selection method [85] which demarcates clusters by constructing Delaunay tessellation of a user-provided radius on a distribution of precipitate atoms. This method works directly on point cloud data instead of homogenized space, thus preventing loss of information. However, this method assumes that the Delaunay radii of the cells follow Poisson's distribution and requires a user input of the Delaunay radius. Our framework, detailed in this paper, can seamlessly integrate with the work of Lefebvre [85] – as GraPTop makes no assumptions on the distribution of atoms – in providing a data-driven Delaunay radius.

4.3 Materials and Methods

A schematic outline of the framework is shown in Fig. 4.2. The framework consists of three stages: The first stage (pre-processing) converts the point cloud data (atomic positions and chemistry information of all the atoms in the region of interest) into a undirected, weighted, and labeled

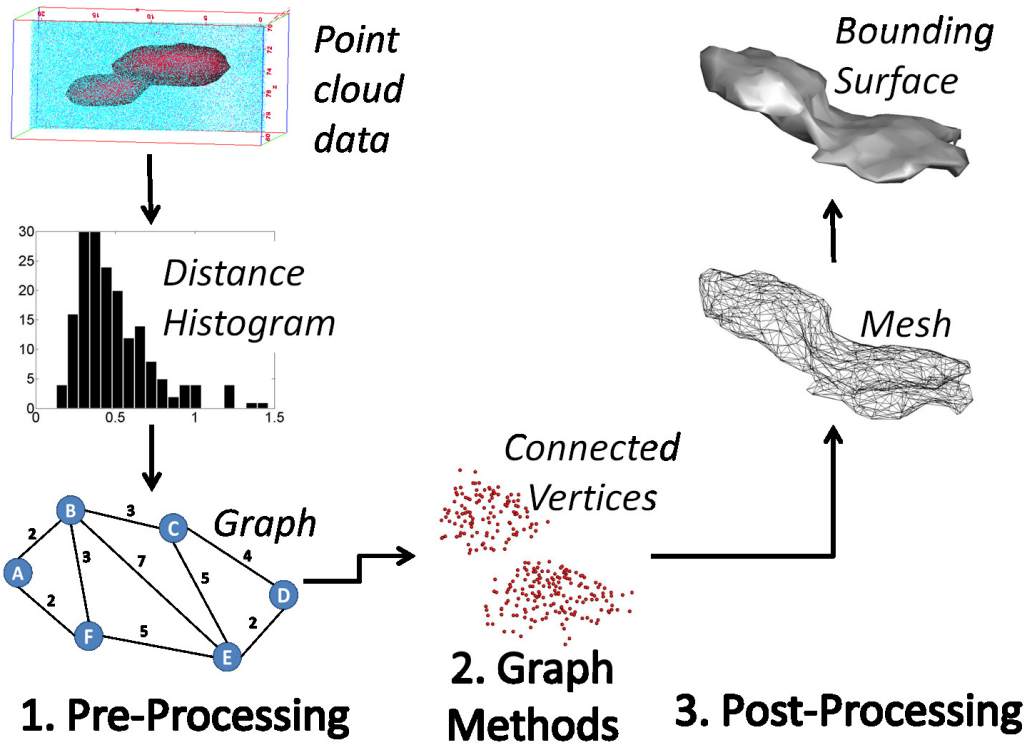

Figure 4.2 Outline of the graph based framework, GraPTop

graph. The second stage deploys graph algorithms to isolate the atoms that constitute the precipitates. The third stage (post-processing) constructs a bounded surface using tessellation and surface mesh generation methods.

The rest of this section details the algorithms in each stage of the framework. The input to the framework is the point cloud data from the Atom Probe given as a set of x, y, z coordinates and mass-to-charge state ratio (m/q) of each of n atoms. We assume that the input is given as a matrix, $[X]$, of size $n \times 4$.

4.3.1 Pre-processing – Converting the point-cloud data into a Graph

The pre-processing stage constitutes of constructing an equivalent graph G from the given atomic position and chemistry information stored in $[X]$. A graph G consists of a set of vertices, V connected with a set of edges E. A label is associated with each vertex and is stored in L. Finally, a non-negative number, called weight, is associated with each edge and is stored in W.

These four components describe a graph, G completely and is denoted as $G(V, E, L, W)$. In order to transform the 3D point cloud data into the equivalent graph G, we assign values to each of the data-structures, V, E, L, W.

Vertex, V: Each atom in the data-set becomes a vertex. [1]

Edge, E: An edge joins a pair of vertices. Intuitively, two neighboring atoms (or vertices) should contain an edge. In order to construct edges of the equivalent graph, G, a notion of neighborhood is needed. We define the ϵ-neighborhood of an atom by a ball of radius ϵ around that atom. All atoms that lie within this ϵ-neighborhood are neighbors (of this atom). The radius ϵ can be chosen between zero to infinity. A neighborhood radius of zero yields a set of disconnected vertices while a radius of infinity connects every vertex with an edge to every other vertex in the graph (complete graph). We determine the value of ϵ based on statistical analysis of neighborhood information of the complete point cloud data. The cumulative histogram of the nearest (precipitate) neighbor (NPN) distance is first constructed (see, for example, Fig.4.7). Probability distribution of nearest precipitate neighbor (NPN) distance, and hence the histogram of NPN distance, is decaying in nature as shown in [62, 22]. This implies that the cumulative of the NPN histogram with optimal bin-size is going to be monotonically increasing. Hence, ϵ radius corresponds to the flat region closest to the elbow on cumulative plot represents the minimum neighborhood radius that is required to capture almost all the atoms in the vicinity of any given atom in the point cloud data as its neighbors. A detailed explanation of the procedure is provided in the form of a step-by-step algorithm provided below.

Label, L: Each atom is given a data-label corresponding to its chemical identity. For ease of implementation, we replace alpha-numeric labels with numbers. Functionality of labels is to classify the atoms based on precipitate and solvent atoms. In algorithmic perspective, the rationale behind defining labels is to have the ability to efficiently apply group-specific operations using graph techniques and extract connectivity information.

A step-by-step algorithmic implementation of edge generation is given below. We assume that

[1] The atoms in [X] are usually arrayed in the order in which they are detected by the mass spectrometer. In this analysis, we do not require information on arrival order. This can however be trivially incorporated for future work by defining another label, L_1 with each atom that represents the arrival time.

we are interested in the precipitate of one particular type of element and (without loss of generality) label the atoms of that element as 1 and the rest of the atoms as 0.

1. **Distance Calculation**: For each atom (of label type 1), find the nearest (precipitate) neighbor distance to an atom of the same label type. Store this data in a set D_{pr}.

2. **Optimal Bin Size**: Compute the optimal bin, h_{opt}, width required to construct the histogram using Silverman's formula: $h_{opt} = 0.9m/n_{pr}^{1/5}$
   $where\ \ m = min(\sqrt{Var(D_{pr})}, IQR(D_{pr})/1.349)$,
   where n_{pr} is the number of observations and D_{pr} is the set of observations, and IQR is the inter quartile range. This formula yields a good estimate of optimal window width and, hence, the bin size to be considered for a given histogram.

3. **Histogram Construction**: Construct a histogram of Nearest Precipitate Neighbor (NPN) distance using the optimal window width h_{opt}.

4. **Cumulative Histogram Construction**: Compute the cumulative of the NPN histogram

5. **Optimal Neighborhood Determination**: Consider the region of the neighborhood in the cumulative of NPN where $\epsilon > 0.554\frac{1}{\sqrt[3]{\rho}}$ [62, 22] and pick the minimum ϵ value in that region where differential to the cumulative reaches zero (ρ here is the number density or number of particles per unit volume). This forms our optimal neighborhood distance ϵ_{opt}. The reason being, this choice of neighborhood defines a strict boundary by casting the outliers out while capturing maximum amount of information about the cluster. This also justifies the term 'optimal' of the optimal neighborhood.

6. **Edge determination**: For every pair of atoms (vertex pairs in the graph), construct an edge between them if they are within ϵ_{opt} distance to each other.

Weight, W: Weights are positive real numbers associated with each edge. The edge weight, W is the (euclidean) distance between the two vertices. Note, only vertices that are neighbors have an edge between them and hence a weight associated with that edge.

These five steps convert the point cloud data, $[X]$ into a labeled, weighted graph, $G(V, E, L, W)$

4.3.2 Graph Methods: Extracting precipitate properties from the Graph

In a multi-component alloy system, solvent and precipitates form distinct domains of various sizes and shapes. It is of interest to identify and quantify these sub-domains in the data. These sub-domains correspond to sets of 'connected' groups of vertices (of the same label) in the equivalent graph, G. We utilize efficient computation of the connected components of a graph in conjunction with filtering of the graph to identify and quantify these precipitates. Graph filtering involves virtual masking of edges to retain only those edges satisfying specific properties. Specifically, the filtering step consists of virtually masking edges between vertices of different labels (i.e. different chemical identity). This is followed by a simple enumeration step that counts the number of connected components (see Fig. 4.3). The full process is accomplished using a simple depth-first-search (DFS) algorithm on the graph. A salient feature of this stage is that the complexity of the DFS algorithm is linear ($\mathcal{O}(n)$), resulting in a highly scalable algorithm. [2] As the outcome of this algorithm, each vertex of the graph has an assigned index of the corresponding component. Fig. 4.3 illustrates a simple example with the three distinct connected components (precipitates) circled.

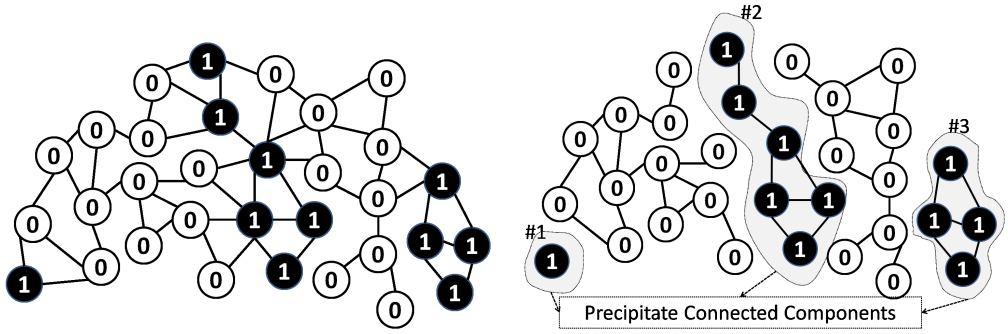

Figure 4.3 (a) Illustration of a graph built using all atoms in a region of interest of APT dataset with black atoms(label=1) representing the precipitate. (b) Three connected components of precipitate identified in the graph

[2] From an algorithmic perspective, this is the only step which works on the full data (both solvent and precipitate) while the pre-processing and post-processing steps work only on precipitate data

4.3.3 Post Processing: Rendering bounded surfaces from connected component data

The connected component information is subsequently used to construct a smooth bounding surface and further processed to extract precipitate topology characteristics like area, volume, nonconvexity of the precipitates. We use local Delaunay tessellation [32] to construct a solid mesh of the precipitate and subsequently render the surface. Delaunay tessellation is a mathematical tool for reconstructing a volume-covering from a discrete point cloud data. Following the constructing of the precipitate shape, we extract a variety of topological information from the shape.

We enumerate these steps below:

1. Each connected component represents a unique precipitate. Store the x, y, z coordinates of atoms belonging to each connected component in a matrix, $[X_{pr}]$

2. Construct Delaunay tessellation [32] with a radius equal to the neighborhood radius ($R_{delaunay} = \epsilon_{opt}$) over the above extracted set of points $[X_{pr}]$ using the Quickhull algorithm [16]. Note that the choice of $R_{delaunay} = \epsilon_{opt}$ is optimal based on the fact that neighborhood radius (ϵ_{opt}) is the minimum radius required to obtain all the neighborhood information. Choosing $R_{delaunay} < \epsilon_{opt}$ will lead to isolation of atoms and choosing $R_{delaunay} > \epsilon_{opt}$ will lead to loss of non-convexity information of the precipitate. The output from this step is a list of sets of three vertices which form a unique triangle (or face) which form tetrahedra (the Delauny tessellation).

3. Extract surface elements from volume elements: The Delauny tessellation meshes the entire volume of the precipitate. We are interested in only rendering the outer surface. We easily reduce this data set by extracting the surface tessellation. This is done by removing the faces (triangles) occurring twice (these form part of the inner elements).

4. Using this mesh (tessellation) information, a surface is rendered and characteristics like area, volume and non-convexity is extracted as follows:

   (a) The surface area of the precipitate is computed using Heron's formula:

$$A_i = \sqrt{s * (s-a) * (s-b) * (s-c)} \tag{4.1}$$

where s is the semiperimiter of each triangle in the Delaunay tessellation. i varies from 1 to δ where δ is the number of surface triangles. The total surface area is the sum of the areas of individual triangles:

$Area_{total} = \sum_{i=1}^{\delta} Area_i$

(b) Volume is computed by summing up the volumes of each individual (tetrahedral or volume) element of the tessellation. Volume of a tetrahedron is given by:

$\mathcal{V}_j = \frac{1}{3} * A_j * H_j$ where A_j is the area of the base triangle and H_j is the height of the tetrahedron with respect to the base triangle. The total volume is given by:

$\mathcal{V}_{total} = \sum_{j=1}^{n_t} \mathcal{V}_j$ where n_t is the number of tetrahedral elements.

(c) A measure of non-convexity is estimated by computing the percentage difference the actual rendered surface and the convex surface that circumscribe the precipitate: $\frac{\mathcal{V}_{convex} - \mathcal{V}_{actual}}{\mathcal{V}_{actual}}$ where Vol_{convex} is the volume of the convex hull circumscribing precipitate. This convex hull is constructed by using a delaunay radius of infinity ($R_{delaunay} = \infty$).

Fig. 4.4 illustrates a representative set of output for a simple, model example. Note that the connected component with single isolated atom is discarded at this post-processing stage.

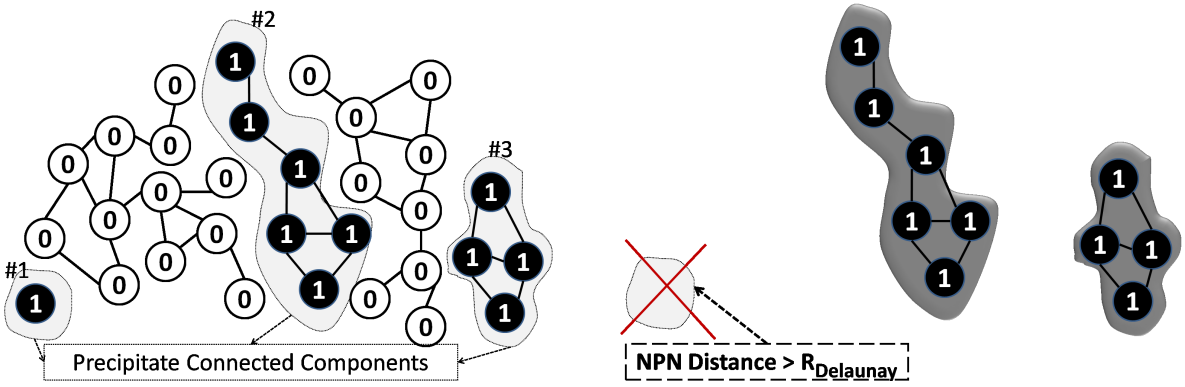

Figure 4.4 (a) Connected component vertices (b) 2D surface tessellation along with a rendered surface

These three stages accomplish the conversion of a point cloud APT data into rendered surface describing the precipitate geometry. Fig. 4.5 shows a block diagram of the complete framework.

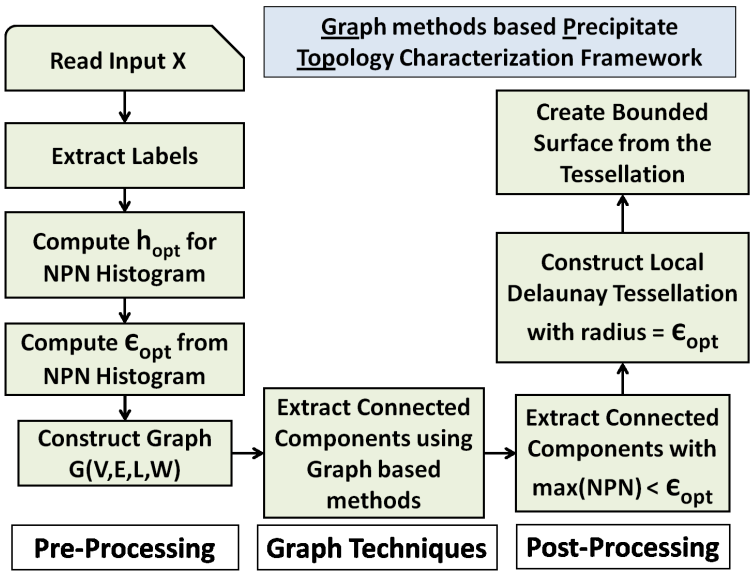

Figure 4.5 Methodology of GraPTop

4.4 Results and Discussion

This section showcases the results obtained by applying GraPTop to three different regions of interest in a point cloud data of Al-Mg-Sc (Aluminium-Magnesium-Scandium) alloy.The composition of the alloy is Al-3.65 Mg-0.566 Sc (at.%). APT images show the presence of Sc rich ($Li_2 - Al_3Sc$) precipitates embedded in an Al-rich solvent. However, information about the volume of the precipitate and the interconnectivity of the precipitates is not clear directly from the APT images. Such information is necessary to understand the kinetic pathways for coagulation of the precipitates and help design the precipitate microstructure in the alloy. The GraPTop methodology uses graph based heuristic-free methods to identify the interconnectivity amongst the precipitates and quantify the size and shape of the precipitates.

4.4.1 Description of the input dataset: Al-Mg-Sc alloy

We consider **three separate regions of interests** in a point cloud of atoms of an Al-Mg-Sc alloyed specimen, and is shown in Fig. 4.6. The input to the framework is a set of x,y,z coordinates and the m/q (mass-to-charge state) ratios of all atoms in each region of interest.

Vertices (V), edges (E), weights (W) and labels (L) for constructing the undirected, weighted,

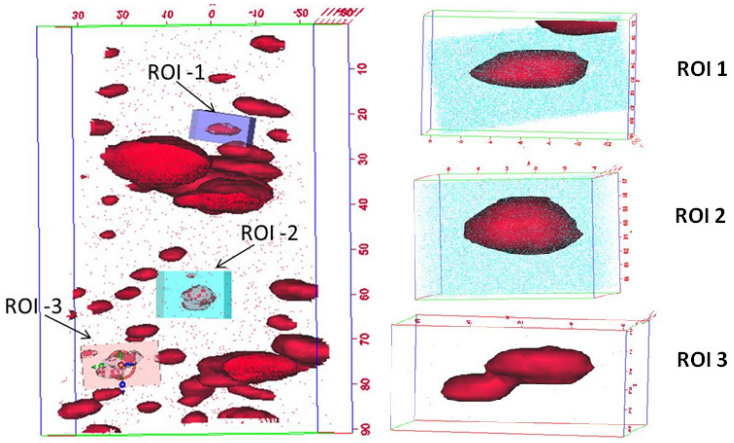

Figure 4.6 3D Atom probe tomograph reconstruction of Al-Mg-Sc alloy, regions of interests: ROI 1,2,3 that form inputs to GraPTop

labeled graph G(V,E,W,L) are first evaluated. We are interested in understanding Scandium precipitate shapes. Consequently, the label for Sc atoms is 1, while the rest are assigned a label 0 Al=0, Mg=0, Sc=1, others=0.

Fig. 4.7- 4.9 represent the histograms (on the left) and cumulative histograms (on the right) for the three regions of interest. The optimal ϵ is chosen from the cumulative histogram as the point where the cumulative value reaches stability (or the slope of the curve diminishes). This point is marked with a circle in the cumulative plots (ϵ_{opt}). In all the 3 cases, there is an abruptly curving elbow in cumulative plots indicating the choice of optimal neighborhood radius (ϵ_{opt}).

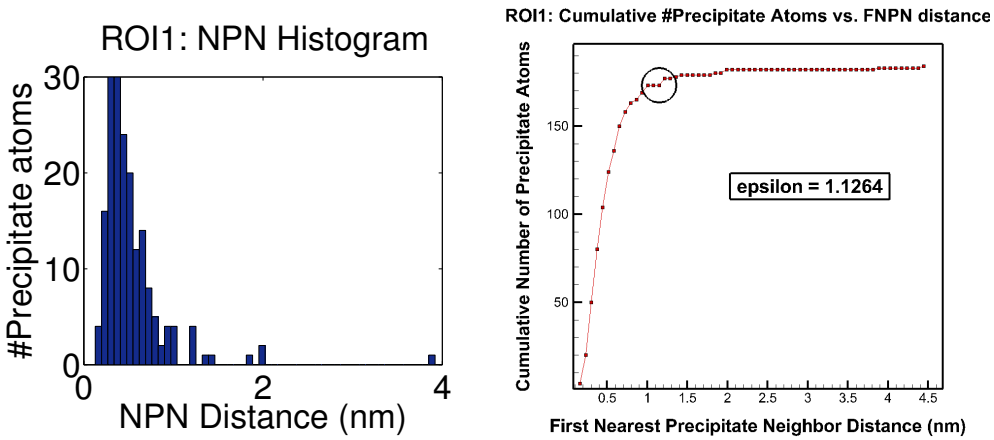

Figure 4.7 (a) ROI 1: Histograms of NPN distance at window width, $h_{opt1} = 0.07$ (b)ROI 1: Cumulative of the histogram

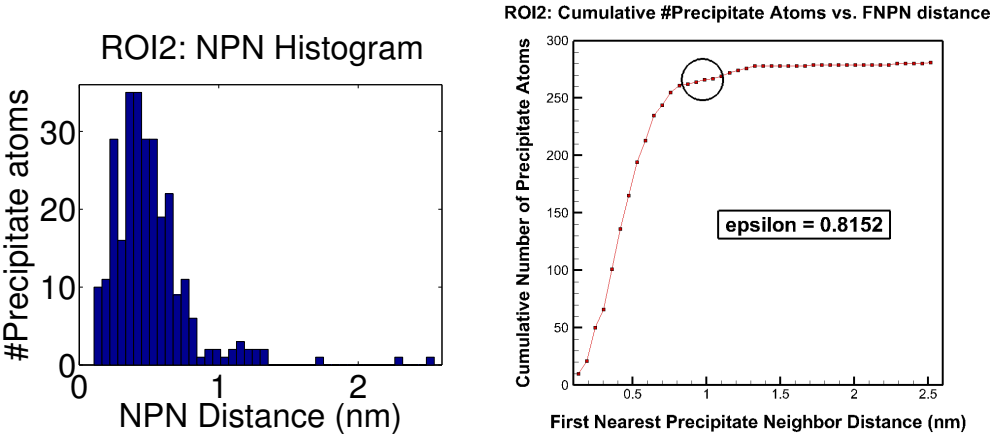

Figure 4.8 (a)ROI 2: Histograms of NPN distance at window width, $h_{opt2} = 0.057$ (b)ROI 2: Cumulative of the histogram

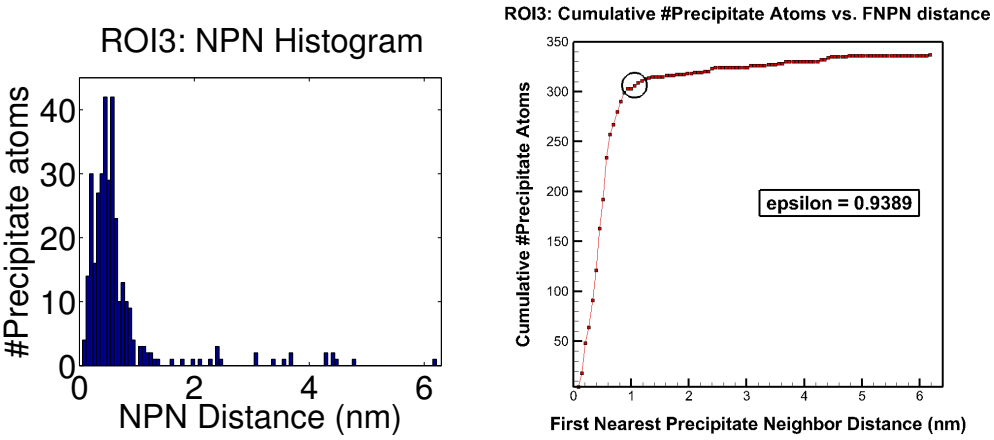

Figure 4.9 (a)ROI 3: Histograms of NPN distance at window width, $h_{opt3} = 0.061$ (b)ROI 3: Cumulative of the histogram

The undirected, weighted, labeled graph G(V,E,W,L) is constructed and the connected components are extracted. Tessellation of the connected components yields the output shown in Fig. 4.10-4.12. These figures show the qualitative output of the GraPTop framework. Adjacent to each of these figures, the corresponding convex hull circumscribing the precipitate is also shown. By computing and comparing the volumes of the convex hull and the original surface topology, a measure of non-convexity of the original surface can be obtained. The change in volume estimated by convex and non-convex surfaces gives a measure of non-convexity of the precipitate. Convex and non-convex volumes as well as other measures like optimal epsilon and optimal window width are

tabulated in Table 4.1. These values indicate that ROI 3 has the largest non-convexity. This information is very valuable to understand the kinetic pathways for coagulation of the precipitates.

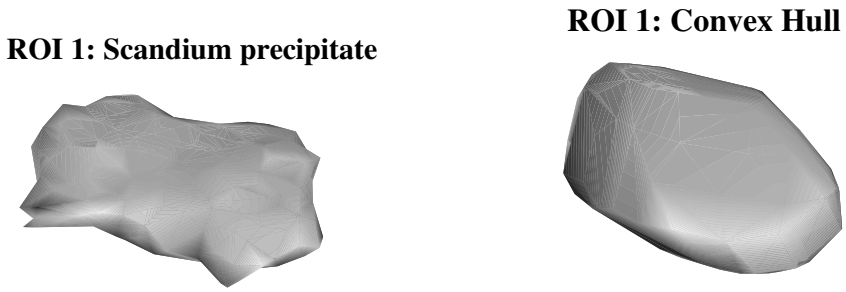

Figure 4.10 ROI 1: (a) Non-convex surface of precipitate. (b)Convex hull of precipitate

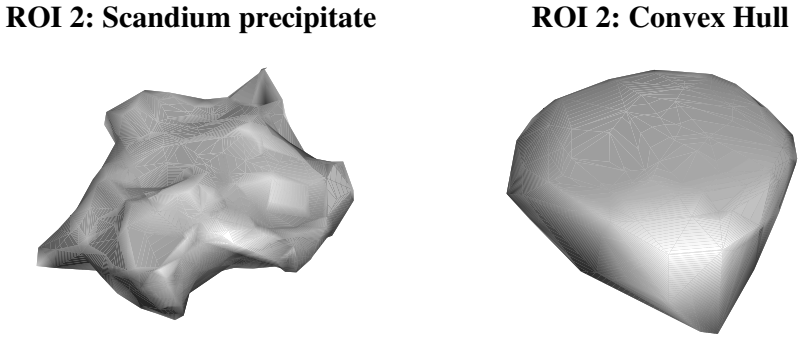

Figure 4.11 ROI 2: (a) Non-convex surface of precipitate. (b)Convex hull of precipitate

This measure of non-convex volume provides a meaningful parameter that is gained through this technique. For example, by identifying the large change in volume for ROI 3, we are able to identify it as an example of a cluster that is formed by initial stage coagulation of two clusters. Quantitative evidence of this stage is not obtained through typical analyses. An additional advantage of the GraPTop technique is the more defined procedure for defining precipitates, as opposed to typical

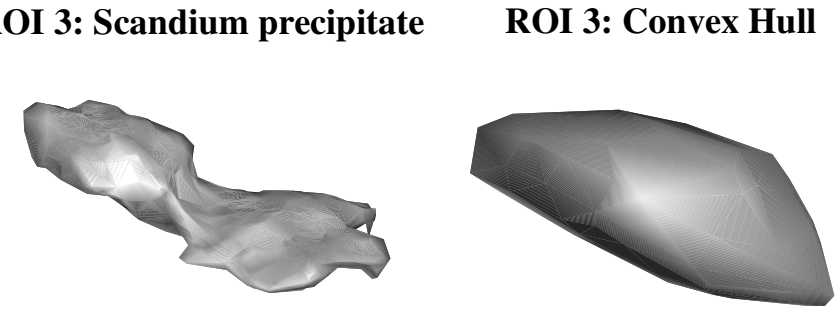

Figure 4.12 ROI 3: (a) Non-convex surface of precipitate. (b)Convex hull of precipitate

approaches where the primary effort is in defining concentration thresholds and voxel sizes which provide an image matching the assumed shape of the precipitate. By defining precipitates based on a parameter that has a clear guideline for selection (ϵ_{opt}), a single measure of area and volume is determined. This provides a significant advantage over the typical approach of reporting values for multiple concentration thresholds, as shown in Figure 4.13, where the definition of precipitate is based on visual bias.This figure demonstrates a standard approach for defining precipitates, where the threshold is defined based on visual bias, resulting in arbitrary measurements of precipitate size and volume. The GraPTop approach removes this arbitrariness from defining the precipitates.

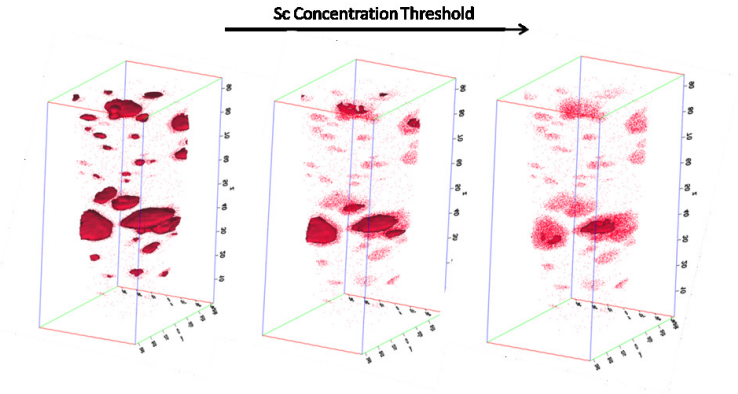

Figure 4.13 Concentration isosurfaces of precipitates as a function of Sc concentration threshold value

Table 4.1 Quantitative Results for ROI 1,2,3

Parameter	**ROI 1**	**ROI 2**	**ROI 3**
ϵ_{opt}	1.1264	0.8512	0.9389
Area(nm^2)	76.0146	99.2236	168.0230
Convex Volume (nm^3)	44.7701	86.1100	962.5300
Non-convex Volume (nm^3)	25.9072	41.9127	58.3814
% Change in Volume	72.8095	105.4509	1548.6930

4.5 Conclusion

In this paper,we formulate the problem of characterization of the precipitates from point cloud APT data as a graph problem. We present a robust, heuristic-free graph-theoretic methodology to solve the formulated problem and provide an implementation of it along with the results obtained by applying the GraPTop framework to three APT point cloud datasets of Al-Mg-Sc alloy. Our framework is robust due to its independence from heuristics like concentration level. We envision applying this framework on an array of datasets obtained from atom probe reconstruction where each dataset is prepared by regulated variation in the process of fabrication. This process of parametric study of the space can give insights into the relationship between the topology of the precipitates and the fabrication process. We are currently also extending and integrating this framework to analyze the homological properties [74] of precipitates. We are also currently working on a mathematical formulation based on random graphs to extend the current framework to account for epistemic uncertainties. This will enable us to provide probabilistic bounds on precipitate descriptors due to the inherent uncertainty in APT measurements.

Acknowledgments

This research was supported in part by the National Science Foundation through XSEDE resources provided by TACC under grant number TG-CTS110007, and supported in part by NSF PHY-0941576, and NSF-0917202.

CHAPTER 5. CONCLUSION

Large materials data generated using high-throughput experimentation formed a rich source of information to establish process-structure-property relationships. This necessitated the development of several mathematical models and scalable techniques to analyze the data.

In the first part of the thesis, we have detailed a mathematical framework of selected nonlinear dimensionality reduction techniques for constructing reduced order models of complicated datasets and discussed key questions involved in data selection. During that process we have also introduced the basic principles behind data dimensionality reduction and illustrated their use with the help of example apatite dataset in materials science using both linear and non-linear methods [1]. Another significant contribution of this paper is that we also describe a rigorous technique (based on graph-theoretic analysis) to estimate the optimal dimensionality of the low-dimensional (or parametric) representation. These techniques are packaged into a modular, computational scalable software framework with a graphical user interface - Scalable Extensible Toolkit for Dimensionality Reduction (SETDiR). This interface helps to separate out the mathematics and computational aspects from the scientific applications, thus significantly enhancing utility of DR techniques to the scientific community.

In order to cater to the needs of larger datasets we illustrated a systematic analysis of spectral dimensionality reduction techniques in the second part of the thesis. We also recast these techniques into a unified view that can be exploited by dimensionality reduction algorithm designers. We subsequently identified the common computational building blocks required to implement a spectral dimensionality reduction method. We used this insight to design and implement a parallel

[1] A comprehensive catalogue of nonlinear dimensionality reduction techniques along with the mathematical prerequisites for understanding dimensionality reduction could be found at: [84]

framework for dimensionality reduction that can handle large datasets, and scales to thousands of processors. We demonstrated the capability and scalability of this framework on several test datasets. We finally showcased the applicability and potential of the framework towards unravelling complex process-morphology relationships in the manufacture of plastic solar cells.

In the third part of the thesis, we formulate the problem of characterization of the precipitates from point cloud APT data as a graph problem. We present a robust, heuristic-free graph-theoretic methodology as well as an implementation to solve the formulated problem. The applicability of the framework was illustrated on 3 different regions of Scandium precipitate in Al-Mg-Sc alloy. Interesting quantitative measures of area, volume and non-convexity were extracted, which can be used to understand parameters like degree of kinetic coagulation of the precipitates in a heterogenous mixture.

5.1 Future Work

Dimensionality Reduction (DR) techniques have proved to be quite successful on a set of microstructure evolution data in image (or pixel) space in extracting process-structure-property relationships. We are currently applying DR to a set of microstructures defined, not in image space, but in topology space (with each high dimension axis representing one topological property like: connectivity, domain-size, interfacial area of a binary microstructure). These topological properties of a given binary microstructure are extracted using a Graph-based Structure Property Investigator (GraSPI) [151]. We anticipate to map a much more efficient low-dimensional representation with this novel metric, and extract interesting quantitative correlations between the process variables, microstructures and specific topological properties. Furthermore, applying DR techniques can also give us insights into an optimal quantitative representation of a given microstructure. Another interesting problem in the pipeline stems from the fact that a change in the choice of solvent, solvent properties like evaporation rate can affect the structure (or nanomorphology) and hence the performance of organic solar cells [5, 54]. We plan to apply our in-house DR framework to a set of potential solvents described in property space along with the performance variables in order to

establish process-structure-property relationship.

The current version of parallel DR framework (PaDRe) has a capability of solving thousands of points in a million dimensional space. However, due to several calls of large, dense, matrix-matrix multiplications ($O(n^3)$), as the problem size increases it begins to grow extremely slow. To overcome this difficulty, we are currently implementing matrix-matrix multiplication routines from BLAS [38] package in our framework. We anticipate a significant performance difference not just with respect to the DR framework but also with respect to the power-iteration based eigensolver since majority of the latter solver involves performing matrix-vector multiplications.

As a part of future work, we envision applying the GraPTop framework on an array of datasets obtained from atom probe reconstruction where each dataset is prepared by regulated variation in the process of fabrication. This process of parametric study of the space can give insights into the relationship between the topology of the precipitates and the fabrication process. We are currently also extending and integrating this framework to analyze the homological properties [74] of precipitates. We are also currently working on a mathematical formulation based on random graphs to extend the current framework to account for epistemic uncertainties. This will enable us to provide probabilistic bounds on precipitate descriptors due to the inherent uncertainty in APT measurements.

BIBLIOGRAPHY

[1] A., B., Gilbert, J., and Budak, C. (2010). Solving path problems on the gpu. *Parallel Computing*, 36:241 – 253. Parallel Matrix Algorithms and Applications.

[2] Andersson, E. and Ekström, P. (2004). Investigating google's pagerank algorithm. Technical report, UPPSALA UNIVERSITY.

[3] Andrecut, M. (2009). Parallel gpu implementation of iterative pca algorithms. *Journal of Computational Biology*, 16(11):1593–1599.

[4] Anisimov, V., Aryasetiawan, F., and Lichtenstein, A. (1997). First-principles calculations of the electronic structure and spectra of strongly correlated systems: the lda+ u method. *Journal of Physics: Condensed Matter*, 9(4):767.

[5] Arias, A., MacKenzie, J., Stevenson, R., Halls, J., Inbasekaran, M., Woo, E., Richards, D., and Friend, R. (2001). Photovoltaic performance and morphology of polyfluorene blends: a combined microscopic and photovoltaic investigation. *Macromolecules*, 34(17):6005–6013.

[6] Ashcroft, N. (1966). Electron-ion pseudopotentials in metals. *Physics Letters*, 23(1):48–50.

[7] Balachandran, P. V. (2011). *Statistical learning for chemical crystallography*. PhD thesis, Iowa State University.

[8] Balachandran, P. V. and Rajan, K. (2012). Structure maps for ${A^I}_4 A^I I_6 (BO_4)_6 X_2$ apatite compounds *via* data mining. *Acta Crystallographica Section B*, 68(1):24–33.

[9] Balachandran, P. V., Samudrala, S., Ganapathysubramanian, B., and Rajan, K. (2013). Comparative study of data dimensionality reduction methods for the discovery of materials chemistries for toxicity immobilization. "pre-print".

[10] Balay, S., Brown, J., Buschelman, K., Eijkhout, V., Gropp, W., Kaushik, D., Knepley, M., McInnes, L., Smith, B., and Zhang, H. (2010). *PETSc Users Manual*.

[11] Bardeen, J. and Pines, D. (1955). Electron-phonon interaction in metals. *Physical Review*, 99(4):1140.

[12] Beardwood, J., Halton, J. H., and Hammersley, J. M. (1959). The shortest path through many points. *Mathematical Proceedings of the Cambridge Philosophical Society*, 55:299–327.

[13] Belkin, M. and Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396.

[14] Bergman, S. (1950). The kernel function and conformal mapping. *American Mathematical Society*.

[15] Bernstein, M., De Silva, V., Langford, J., and Tenenbaum, J. (2000). Graph approximations to geodesics on embedded manifolds. Technical report, Technical report, Department of Psychology, Stanford University.

[16] Bradford Barber, C., Dobkin, D., and Huhdanpaa, H. (1996). The quickhull algorithm for convex hulls. *ACM TRANSACTIONS ON MATHEMATICAL SOFTWARE*, 22(4):469–483.

[17] Brasca, R., Vergara, L. I., Passeggi, M. C. G., and Ferrona, J. (2007). Chemical changes of titanium and titanium dioxide under electron bombardment. *Materials Research*, 10:283 – 288.

[18] Breitmoser, E. and Sunderland, A. (2003). An overview of eigensolvers for hpcx. Technical report, HPCx Consortium.

[19] Brideau, C., Gunter, B., Pikounis, B., and Liaw, A. (2003). Improved statistical methods for hit selection in high-throughput screening. *Journal of Biomolecular Screening*, 8(6):634–647.

[20] Cattell, R. (1944). A note on correlation clusters and cluster search methods. *Psychometrika*, 9(3):169–184.

[21] Ceguerra, A., Moody, M., Stephenson, L., Marceau, R., and Ringer, S. (2010). A three-dimensional markov field approach for the analysis of atomic clustering in atom probe data. *Philisophical Magazine*, 90:1657–1683.

[22] Chandrasekhar, S. (1943). Stochastic problems in physics and astronomy. *Rev. Mod. Phys.*, 15:1–89.

[23] Chawla, N., Ganesh, V., and Wunsch, B. (2004). Three-dimensional (3d) microstructure visualization and finite element modeling of the mechanical behavior of sic particle reinforced aluminum composites. *Scripta Materialia*, 51(2):161 – 165.

[24] Chen, L., Hong, Z., Li, G., and Yang, Y. (2009). Recent progress in polymer solar cells: Manipulation of polymer:fullerene morphology and the formation of efficient inverted polymer solar cells. *Advanced Materials*, 21(14-15):1434–1449.

[25] Choi, J., Demmel, J., Dhillon, I., Dongarra, J., Ostrouchov, S., Petitet, A., Stanley, K., Walker, D., and Whaley, R. (1996). Scalapack: a portable linear algebra library for distributed memory computers - design issues and performance. *Computer Physics Communications*, 97(1-2):1 – 15. High-Performance Computing in Science.

[26] Coffey, D. C. and Ginger, D. (2005). Patterning phase separation in polymer films with dip-pen nanolithography. *J. Am. Chem. Soc*, 127:4564–4565.

[27] Cohen, M. (1982). Pseudopotentials and total energy calculations. *Physica Scripta*, 1982(T1):5.

[28] Crauser, A., Mehlhorn, K., Meyer, U., and Sanders, P. (1998). A parallelization of dijkstra's shortest path algorithm. In *IN PROC. 23RD MFCS'98, LECTURE NOTES IN COMPUTER SCIENCE*, pages 722–731. Springer.

[29] Curtarolo, S., Morgan, D., Persson, K., Rodgers, J., and Ceder, G. (2003). Predicting crystal structures with data mining of quantum calculations. *Phys. Rev. Lett.*, 91:135503.

[30] D., K., Callaghan, S., Harkness, R., Jha, S., Kurowski, K., Manos, S., Pamidighantam, S., Marlon, P., Beth, P., Song, C., and Towns, J. (2010). Science on the teragrid. *Computational Methods in Science and Technology*, pages 87–97. Special Issue.

[31] Davison, M. (1983). *Multidimensional scaling*. Wiley New York.

[32] de Berg, M., Cheong, O., van Kreveld, M., and Overmars, M. (2008). *Computational Geometry: Algorithms and Applications*. Springer.

[33] De Silva, V. and Tenenbaum, J. (2004). Sparse multidimensional scaling using landmark points. Technical report, Technical report, Stanford University.

[34] Deibel, C., Dyakonov, V., and Brabec, C. (2010). Organic bulk-heterojunction solar cells. *Selected Topics in Quantum Electronics, IEEE Journal of*, 16(6):1517 –1527.

[35] Delling, D., Goldberg, A. V., Nowatzyk, A., and Werneck, R. F. (2012). Phast: Hardware-accelerated shortest path trees. *Journal of Parallel and Distributed Computing*.

[36] Demartines, P. and Hérault, J. (1997). Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. *Neural Networks, IEEE Transactions on*, 8(1):148–154.

[37] Dhillon, I., Parlett, B., and Vömel, C. (2006). The design and implementation of the mrrr algorithm. *ACM Transactions on Mathematical Software*, 32(4):533–560.

[38] Dongarra, J., Du Croz, J., Hammarling, S., and Hanson, R. (1988). An extended set of fortran basic linear algebra subroutines. *ACM Trans. Math. Soft*, 14(1):1–17.

[39] Donoho, D. and Grimes, C. (2003). Hessian eigenmaps: New locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, 100:5591–5596.

[40] Elliott, J. C. (1994). *Structure and chemistry of the apatites and other calcium orthophosphates*, volume 4. Elsevier Amsterdam.

[41] Fang, C., Ahuja, R., and Eriksson, O. (2007). Prediction of max phases, vsic (n= 1, 2), from first-principles theory. *Journal of Applied Physics*, 101:013511.

[42] Fischer, C., Tibbetts, K., Morgan, D., and Ceder, G. (2006). Predicting crystal structure by merging data mining with quantum mechanics. *Nature materials*, 5(8):641–646.

[43] Flora, N., Hamilton, K., Schaeffer, R., and Yoder, C. (2004). A comparative study of the synthesis of calcium, strontium, barium, cadmium, and lead apatites in aqueous solution. *Synthesis and Reactivity in Inorganic and Metal-organic Chemistry*, 34(3):503–521.

[44] Floyd, R. W. (1962). Algorithm 97: Shortest path. *Commun. ACM*, 5(6):345–.

[45] Fontanini, A., Olsen, M., and Ganapathysubramanian, B. (2011). Thermal comparison between ceiling diffusers and fabric ductwork diffusers for green buildings. *Energy and Buildings*.

[46] Galtrey, M., Oliver, R., Kappers, M., Humphreys, C., Clifton, P., Larson, D., Saxey, D., and Cerezo, A. (2008). Three-dimensional atom probe analysis of green- and blue-emitting in_xga_{1-x}n/gan multiple quantum well structures. *Journal of Applied Physics*, 104(1).

[47] Ganapathysubramanian, B. and Zabaras, N. (2008). A non-linear dimension reduction methodology for generating data-driven stochastic input models. *Journal of Computational Physics*, 227(13):6612 – 6637.

[48] Geiser, B., Kelly, T., Larson, D., Schneir, J., and Roberts, J. (2007). Spatial distribution maps for atom probe tomography. *Microscopy and Microanalysis*, 13:437–447.

[49] Georges, A., Kotliar, G., Krauth, W., and Rozenberg, M. (1996). Dynamical mean-field theory of strongly correlated fermion systems and the limit of infinite dimensions. *Reviews of Modern Physics*, 68(1):13.

[50] Geuser, F. and Lefebvre, W. (2011). Determination of matrix composition based on solute-solute nearest neighbor distances in atom probe tomography. *Microscopy Research and Technique*, 74:257–263.

[51] Golub, G. and Van Loan, C. (1996). *Matrix Computations*. The John Hopkins University Press.

[52] Grassberger, P. and Procaccia, I. (1983). Measuring the strangeness of strange attractors. *Physica D: Nonlinear Phenomena*, 9(1-2):189 – 208.

[53] Greeley, J., Jaramillo, T., Bonde, J., Chorkendorff, I., and Nørskov, J. (2006). Computational high-throughput screening of electrocatalytic materials for hydrogen evolution. *Nature materials*, 5(11):909–913.

[54] Gunes, S., Neugebauer, H., and Sariciftci, N. (2007). Conjugated polymer-based organic solar cells. *Chemical Reviews-Columbus*, 107(4):1324–1338.

[55] Guo, Q., Rajewski, D., Takle, E., and Ganapathysubramanian, B. (2012). Constructing low-dimensional stochastic wind models from meteorology data. in-preparation.

[56] Hariharan, B. and Aluru, S. (2005). Efficient parallel algorithms and software for compressed octrees with applications to hierarchical methods. *Parallel Computing*, 31(3-4):311 – 331.

[57] Harrison, W. (1966). *Pseudopotentials in the theory of metals*. Frontiers in physics. W.A. Benjamin.

[58] Hartigan, J. A. (1975). *Clustering algorithms*. John Wiley & Sons, Inc.

[59] Hellman, O., Vandenbroucke, J., Blatz du Rivage, J., and N., S. D. (2002). Application software for data analysis for three-dimensional atom probe microscopy. *Materials Science and Engineering A*, 327:29–33.

[60] Hendrickson, B. and Plimpton, S. (1995). Parallel many-body simulations without all-to-all communication. *Journal of Parallel and Distributed Computing*, 27(1):15 – 25.

[61] Hernandez, V., Roman, J., and Vidal, V. (2005). Slepc: A scalable and flexible toolkit for the solution of eigenvalue problems. *ACM Trans. Math. Softw.*, 31(3):351–362.

[62] Hertz, P. (1909). ÃIJber den gegenseitigen durchschnittlichen abstand von punkten, die mit bekannter mittlerer dichte im raume angeordnet sind. *Mathematische Annalen*, 67:387–398.

[63] Hohenberg, P. and Kohn, W. (1964). Inhomogeneous electron gas. *Physical Review*, 136(3B):B864.

[64] Hohenberg, P., Kohn, W., and Sham, L. (1990). The beginnings and some thoughts on the future. *Advances in Quantum Chemistry*, 21:7–26.

[65] Hoppe, H. and Sariciftci, N. S. (2006). Morphology of polymer/fullerene bulk heterojunction solar cells. *J. Mater. Chem.*, 16:45–61.

[66] Huang, F., Chen, L., Wu, Z., and Wang, W. (2013). First-principles calculations of equilibrium mg isotope fractionations between garnet, clinopyroxene, orthopyroxene, and olivine: Implications for mg isotope thermometry. *Earth and Planetary Science Letters*, 367:61–70.

[67] Humblet, P. (1991). Another adaptive distributed shortest path algorithm. *Communications, IEEE Transactions on*, 39(6):995–1003.

[68] Hyde, J., Marquis, E., Wilford, K., and Williams, T. (2011). A sensitivity analysis of the maximum separation method for the characterisation of solute clusters. *Ultramicroscopy*, 111(6):440 – 447.

[69] Ingram, S., Munzner, T., and Olano, M. (2009). Glimmer: Multilevel mds on the gpu. *Visualization and Computer Graphics, IEEE Transactions on*, 15(2):249–261.

[70] J., P., B., M., van de Geijn R.A., J.R., H., and Romero, N. (2012). Elemental: A new framework for distributed memory dense matrix computations. forthcoming.

[71] Jandeleit, B., Schaefer, D., Powers, T., Turner, H., and Weinberg, W. (1999). Combinatorial materials science and catalysis. *Angewandte Chemie International Edition*, 38(17):2494–2532.

[72] Jenq, J. and Sahni, S. (1987). All pairs shortest paths on a hypercube multiprocessor. *International Conference on Parallel Processing*, pages 713–716.

[73] K., M., Bader, D., Berry, J., and Crobak, J. (2007). An experimental study of a parallel shortest path algorithm for solving large-scale graph instances.

[74] Kaczynski, T., Mischaikow, K., and Mrozek, M. (2004). *Computational Homology*. Springer.

[75] Katz, G. and Kider, J. J. (2008). All-pairs shortest-paths for large graphs on the gpu. In *Proceedings of the 23rd ACM SIGGRAPH/EUROGRAPHICS symposium on Graphics hardware*, GH '08, pages 47–55, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

[76] Kohonen, T. (2001). *Self-organizing maps*, volume 30. Springer Verlag.

[77] K.R. Muller, K., Mika, S., Ratsch, G., Tsuda, K., and Scholkopf, B. (2001). An introduction to kernel-based learning algorithms. *Neural Networks, IEEE Transactions*, 12(2):181–201.

[78] Krakauer, B. and Seidman, D. (1998). Subnanometer scale study of segregation at grain boundaries in an fe(si) alloy. *Acta Materialia*, 46(17):6145–6161.

[79] Krakauer, B. W. and Seidman, D. N. (1993). Absolute atomic-scale measurements of the gibbsian interfacial excess of solute at internal interfaces. *Phys. Rev. B*, 48:6724–6727.

[80] Kruskal, J. (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27.

[81] Kruskal, J. and Wish, M. (1978). *Multidimensional scaling*. Beverly Hills: Sage Publications.

[82] Lanczos, C. (1950). An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 45(4).

[83] Langer, S. A., Fuller, E. R., J., and Carter, W. C. (2001). Oof: an image-based finite-element analysis of material microstructures. *Computing in Science Engineering*, 3(3):15 –23.

[84] Lee, J. and Verleysen, M. (2008). *Nonlinear Dimensionality Reduction*. Springer.

[85] Lefebvre, W., Philippe, T., and Vurpillot, F. (2011). Application of delaunay tessellation for the characterization of solute-rich clusters in atom probe tomography. *Ultramicroscopy*, 111(3):200 – 206.

[86] Liu, A. and Cohen, M. (1990). Structural properties and electronic structure of low-compressibility materials: β-si_ {3} n_ {4} and hypothetical β-c_ {3} n_ {4}. *Physical Review B*, 41(15):10727.

[87] Liu, Z. K., Chen, L. Q., Raghavan, P., Du, Q., Sofo, J. O., Langer, S. A., and Wolverton, C. (2004). An integrated framework for multi-scale materials simulation and design. *Journal of Computer-Aided Materials Design*, 11:183–199.

[88] Lumley, J. (1967). The structure of inhomogeneous turbulent flows. *Atmospheric Turbulence and Radio Wave Propagation*, pages 166–178.

[89] Marquis, E. and Hyde, J. (2010). Applications of atom-probe tomography to the characterization of solute behavious. *Materials Science and Engineering Reports*, 69:37–62.

[90] Matsunaga, K., Inamori, H., and Murata, H. (2008). Theoretical trend of ion exchange ability with divalent cations in hydroxyapatite. *Phys. Rev. B*, 78:094101.

[91] McVeigh, C. and Liu, W. K. (2008). Linking microstructure and properties through a predictive multiresolution continuum. *Computer Methods in Applied Mechanics and Engineering*, 197(41-42):3268 – 3290. Recent Advances in Computational Study of Nanostructures.

[92] Meerbergen, K., Spence, A., and Roose, D. (1994). Shift-invert and cayley transforms for detection of rightmost eigenvalues of nonsymmetric matrices. *BIT Numerical Mathematics*, 34:409–423. 10.1007/BF01935650.

[93] Mercier, P. H. J., Le Page, Y., Whitfield, P. S., Mitchell, L. D., Davidson, I. J., and White, T. J. (2005). Geometrical parameterization of the crystal chemistry of p63/m apatites: comparison with experimental data and ab initio results. *Acta Crystallographica Section B: Structural Science*, 61(6):635–655.

[94] Meredith, J. C., Smith, A. P., Karim, A., and Amis, E. J. (2000). Combinatorial materials science for polymer thin-film dewetting. *Macromolecules*, 33(26):9747–9756.

[95] Meyer, U. and Sanders, P. (1998). δ-stepping : A parallel single source shortest path algorithm. In *In ESA* 98*: Proceedings of the 6th Annual European Symposium on Algorithms*, pages 393–404. Springer-Verlag.

[96] Miller, M. (1996). *Atom Probe Field Ion Microscopy*. Monographs on the Physics and Chemistry of Materials. Clarendon Press.

[97] Miller, M. (1999). Characterization of the early stages of phase separation by atom probe tomography. *MRS Online Proceedings Library*, 580.

[98] Miller, M. (2000). *Atom Probe Tomography: Analysis at the Atomic Level*. Kluwer Academic / Plenum Publishers.

[99] Morgan, D., Ceder, G., and Curtarolo, S. (2005). High-throughput and data mining with ab initio methods. *Measurement Science and Technology*, 16(1):296.

[100] Morgan, D., Rodgers, J., and Ceder, G. (2003). Automatic construction, implementation and assessment of pettifor maps. *Journal of Physics: Condensed Matter*, 15(25):4361.

[101] Narasimhan, B., Mallapragada, S., and Porter, M. (2007). *Combinatorial Materials Science*. Wiley.

[102] Page, Y. L. (2006). Data mining in and around crystal structure databases. *MRS Bulletin*, 31:991–994.

[103] Park, S., Roy, A., Beaupre, S., Cho, S., Coates, N., Moon, J., Moses, D., Leclerc, M., Lee, K., and Heeger, A. (2009). Bulk heterojunction solar cells with internal quantum efficiency approaching 100%. *Nature Photon.*

[104] Parlett, B. and Poole, W.G., J. (1973). A geometric theory for the qr, lu and power iterations. *SIAM Journal on Numerical Analysis*, 10(2):389–412.

[105] Parr, R. G. (1983). Density functional theory. *Annual Review of Physical Chemistry*, 34(1):631–656.

[106] Pauling, L. (1960). *The nature of the chemical bond and the structure of molecules and crystals: an introduction to modern structural chemistry*, volume 18. Cornell University Press.

[107] Peet, J., Heeger, A., and Bazan, G. C. (2009). Plastic solar cells: Self-assembly of bulk heterojunction nanomaterials by spontaneous phase separation. *Accounts of Chemical Research*, 42(11):1700–1708. PMID: 19569710.

[108] Pramana, S. S., Klooster, W. T., and White, T. J. (2008). A taxonomy of apatite frameworks for the crystal chemical design of fuel cell electrolytes. *Journal of Solid State Chemistry*, 181(8):1717–1722.

[109] Prim, R. (1957). Shortest connection networks and some generalizations. *Bell system technical journal*, 36(6):1389–1401.

[110] Prosa, T., Clifton, P., Zhong, H., Tyagi, A., Shivaraman, A., DenBaars, S., Nakamura, S., and Speck, J. (2011). Atom probe analysis of interfacial abruptness and clustering within a single in[sub x]ga[sub 1 - x]n quantum well device on semipolar (10[overline 1][overline 1]) gan substrate. *Applied Physics Letters*, 98(19).

[111] Rabe, K. M., Phillips, J. C., Villars, P., and Brown, I. D. (1992). Global multinary structural chemistry of stable quasicrystals, high-t_c ferroelectrics, and high-t_c superconductors. *Phys. Rev. B*, 45:7650–7676.

[112] Rajan, K. (2005). Materials informatics. *Materials Today*, 8(10):38 – 45.

[113] Rajan, K., Suh, C., and Mendez, P. F. (2009). Principal component analysis and dimensional analysis as materials informatics tools to reduce dimensionality in materials science and engineering. *Statistical Analysis and Data Mining*, 1(6).

[114] Rennert, T., Totsche, K., Heister, K., Kersten, M., and Thieme, J. (2012). Advanced spectroscopic, microscopic, and tomographic characterization techniques to study biogeochemical interfaces in soil. *Journal of Soils and Sediments*, 12:3–23.

[115] Roweis, S. and Saul, L. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326.

[116] Saad, Y. (1981). Krylov subspace methods for solving large unsymmetric linear systems. *Mathematics of Computation*, 37(155).

[117] Sakellaridi, S., Fang, H., and Saad, Y. (2009). Multilevel linear dimensionality reduction for data analysis using nearest-neighbor graphs. technical report.

[118] Sammon Jr, J. (1969). A nonlinear mapping for data structure analysis. *Computers, IEEE Transactions on*, 100(5):401–409.

[119] Samudrala, S., Balachandran, P., Broderick, S., Rajan, K., and Ganapathysubramanian, B. (2013a). Nonlinear dimensionality reduction techniques for apatites. archive available.

[120] Samudrala, S., Wodo, O., Suram, S., Broderick, S., Rajan, K., and Ganapathysubramanian, B. (2013b). Nonlinear dimensionality reduction techniques for apatites. under-review.

[121] Samudrala, S., Zola, J., Aluru, S., and Ganapathysubramanian, B. (2013c). Parallel framework for dimensionality reduction of large-scale datasets. archive available.

[122] Sarje, A., Zola, J., and Aluru, S. (2011). Accelerating pairwise computations on cell processors. *IEEE Transactions on Parallel and Distributed Systems*, 22:69–77.

[123] Saul, L., Roweis, S., and Singer, Y. (2003). Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, 4:119–155.

[124] Sean, E. S., Christoph, J., Niyazi, S., Franz, P., Thomas, F., and Jan, C. (2001). 2.5% efficient organic plastic solar cells. *Applied Physics Letters*, 78(6):841–843.

[125] Seidman, D., Krakauer, B., and Udler, D. (1994). Atomic scale studies of solute-atom segregation at grain boundaries: Experiments and simulations. *Journal of Physics and Chemistry of Solids*, 55(10):1035–1057.

[126] Shannon, R. D. (1976). Revised effective ionic radii and systematic studies of interatomic distances in halides and chalcogenides. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):751–767.

[127] Shashkov, D., Muller, D., and Seidman, D. (1999). Atomic-scale structure and chemistry of ceramic/metal interfaces–ii. solute segregation at mgo/cu (ag) and cdo/ag (au) interfaces. *Acta Materialia*, 47(15):3953–3963.

[128] Shashkov, D. and Seidman, D. (1996). Atomic-scale studies of silver segregation at mgocu heterophase interfaces. *Applied Surface Science*, 94–95(0):416–421.

[129] Shashkov, D. A. and Seidman, D. N. (1995). Atomic scale studies of segregation at ceramic/metal heterophase interfaces. *Phys. Rev. Lett.*, 75:268–271.

[130] Shepard, R. (1962). The analysis of proximities: Multidimensional scaling with an unknown distance function. i. *Psychometrika*, 27(2):125–140.

[131] Silva, V. and Tenenbaum, J. (2002). Global versus local methods in nonlinear dimensionality reduction. *Advances in neural information processing systems*, 15:705–712.

[132] Stephenson, L., Moody, M., Liddicoat, P., and Ringer, S. (2007). New techniques for the analysis of fine-scaled clustering phenomena within atom probe tomography (apt) data. *Microscopy and Microanalysis*, 13:488–463.

[133] Stephenson, L., Moody, M., and Ringer, S. (2006). Techniques for the analysis of clusters and aggregations within atom probe tomography data. *Microscopy and Microanalysis*, 12:1732–1733.

[134] Takeuchi, I., Lauterbach, J., and Fasolka, M. J. (2005). Combinatorial materials synthesis. *Materials Today*, 8(10):18 – 26.

[135] Talwalkar, A., Kumar, S., and Rowley, H. (2008). Large-scale manifold learning. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE.

[136] Tenenbaum, J., de Silva, V., and Langford, J. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323.

[137] Torgerson, W. (1952). Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419.

[138] Trau, M. and Battersby, B. (2001). Novel colloidal materials for high-throughput screening applications in drug discovery and genomics. *Advanced Materials*, 13(12-13):975–979.

[139] Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85.

[140] Van der Maaten, L., Postma, E., and Van Den Herik, H. (2009). Dimensionality reduction: A comparative review.

[141] van Rietbergen, B., Weinans, H., Huiskes, R., and Odgaard, A. (1995). A new method to determine trabecular bone elastic properties and loading using micromechanical finite-element models. *Journal of Biomechanics*, 28(1):69 – 81.

[142] Vaumousse, D., Cerezo, A., and Warren, P. (2003). A procedure for quantification of precipitate microstructures from three-dimensional atom probe data. *Ultramicroscopy*, 95:215–221.

[143] Vurpillot, F., Da Costa, G., Menand, A., and Blavette, D. (2001). Structural analyses in three-dimensional atom probe: A fourier transform approach. *Journal of Microscopy*, 203:295–302.

[144] Vurpillot, F., de Geuser, F., Da Costa, G., and Blavette, D. (2004). Application of fourier transform and autocorrelation to cluster identification in the three-dimensional atom probe. *Journal of Microscopy*, 216(3):234–40.

[145] Vurpillot, F., Renaud, L., and Blavette, D. (2003). A new step towards the lattice reconstruction in 3dap. *Ultramicroscopy*, 95:223–229.

[146] White, T., Ferraris, C., Kim, J., and Madhavi, S. (2005). Apatite–an adaptive framework structure. *Reviews in mineralogy and geochemistry*, 57(1):307–401.

[147] White, T. J. and Dong, Z. L. (2003). Structural derivation and crystal chemistry of apatites. *Acta Crystallographica Section B: Structural Science*, 59(1):1–16.

[148] Wilkinson, J. (1965). *The Algebraic Eigenvalue Problem*. Clarendon Press, Oxford.

[149] Wodo, O., Tirthapura, S., Chaudhary, S., and Ganapathysubramanian, B. (2012a). Computational characterization of bulk heterojunction nanomorphology. *Journal of Applied Physics*.

[150] Wodo, O., Tirthapura, S., Chaudhary, S., and Ganapathysubramanian, B. (2012b). A graph-based formulation for computational characterization of bulk heterojunction morphology. *Organic Electronics*, 13(6):1105 – 1113.

[151] Wodo, O., Tirthapura, S., Chaudhary, S., and Ganapathysubramanian, B. (2012c). A novel graph based formulation for characterizing morphology with application to organic solar cells. *Organic Electronics*, pages 1105–1113.

[152] Xie, Y., Hu, H., and Ganapathysubramanian, B. (2011a). Phase transitions in vortex shedding in the wake of a heated circular cylinder at low reynolds number. *2011 Graduate Symposium*.

[153] Xie, Y., Hu, H., and Ganapathysubramanian, B. (2011b). Vortex dynamics in flow past a heated cylinder.

[154] Yang, M. (2002). Face recognition using extended isomap. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 2, pages II–117. IEEE.

[155] Yeh, T., Chen, T., Chen, Y., and Shih, W. (2010). Efficient parallel algorithm for nonlinear dimensionality reduction on gpu. In *Granular Computing (GrC), 2010 IEEE International Conference on*, pages 592–597. IEEE.

[156] Young, G. (1939). Factor analysis and the index of clustering. *Psychometrika*, 4(3):201–207.

[157] Yue, Z., Chen, S., and Tham, L. (2003). Finite element modeling of geomaterials using digital image processing. *Computers and Geotechnics*, 30(5):375 – 397.

[158] Zabaras, N., Sundararaghavan, V., and Sankaran, S. (2006). An information-theoretic approach for obtaining property pdfs from macro specifications of microstructural variability. *TMS letters*, 3:1–2.

[159] Zhang, J., Li, S., and Wang, J. (2004). Nearest manifold approach for face recognition. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pages 223–228. IEEE.

[160] Zhang, X. (2010). Assessing the size of gene or rnai effects in multifactor high-throughput experiments. *Pharmacogenomics*, 11(2):199–213.

[161] Zhong, F., Capson, D., and Schuurman, D. (2008). Parallel architecture for pca image feature detection using fpga. In *Electrical and Computer Engineering, 2008. CCECE 2008. Canadian Conference on*, pages 001341–001344. IEEE.