University of Wisconsin Milwaukee
UWM Digital Commons

Theses and Dissertations

December 2013

Extracting the Structure and Conformations of Biological Entities from Large Datasets

Ali Dashti
University of Wisconsin-Milwaukee

Follow this and additional works at: https://dc.uwm.edu/etd

Part of the Atomic, Molecular and Optical Physics Commons, Biomedical Engineering and Bioengineering Commons, and the Computer Sciences Commons

Recommended Citation

Dashti, Ali, "Extracting the Structure and Conformations of Biological Entities from Large Datasets" (2013). *Theses and Dissertations*. 348.
https://dc.uwm.edu/etd/348

This Dissertation is brought to you for free and open access by UWM Digital Commons. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of UWM Digital Commons. For more information, please contact open-access@uwm.edu.

www.manaraa.com

EXTRACTING THE STRUCTURE AND CONFORMATIONS OF BIOLOGICAL ENTITIES FROM LARGE DATASETS

by

Ali Dashti

A Dissertation Submitted in
Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
in Biomedical and Health Informatics

at
The University of Wisconsin-Milwaukee
December 2013

www.manaraa.com

ABSTRACT

EXTRACTING THE STRUCTURE AND CONFORMATIONS OF BIOLOGICAL ENTITIES FROM LARGE DATASETS

by

Ali Dashti

The University of Wisconsin-Milwaukee, 2013
Under the Supervision of Professor Roshan M. D'Souza

In biology, structure determines function, which often proceeds via changes in conformation. Efficient means for determining structure exist, but mapping conformations continue to present a serious challenge. Single-particles approaches, such as cryogenic electron microscopy (cryo-EM) and emerging “diffract & destroy” X-ray techniques are, in principle, ideally positioned to overcome these challenges. But the algorithmic ability to extract information from large heterogeneous datasets consisting of “unsorted” snapshots - each emanating from an unknown orientation of an object in an unknown conformation - remains elusive.

It is the objective of this thesis to describe and validate a powerful suite of manifold-based algorithms able to extract structural and conformational information from large datasets. These computationally efficient algorithms offer a new approach to determining the structure and conformations of viruses and macromolecules.

After an introduction, we demonstrate a distributed, exact *k*-Nearest Neighbor Graph (*k*-NNG) construction method, in order to establish a firm algorithmic basis for manifold-

المنارة للاستشارات

www.manaraa.com

based analysis. The proposed algorithm uses Graphics Processing Units (GPUs) and exploits multiple levels of parallelism in distributed computational environment and it is scalable for different cluster sizes, with each compute node in the cluster containing multiple GPUs.

Next, we present applications of manifold-based analysis in determining structure and conformational variability. Using the Diffusion Map algorithm, a new approach is presented, which is capable of determining structure of symmetric objects, such as viruses, to $1/100_{th}$ of the object diameter, using low-signal diffraction snapshots. This is demonstrated by means of a successful 3D reconstruction of the Satellite Tobacco Necrosis Virus (STNV) to atomic resolution from simulated diffraction snapshots with and without noise.

We next present a new approach for determining discrete conformational changes of the enzyme Adenylate kinase (ADK) from very large datasets of up to 20 million snapshots, each with $\sim 10^4$ pixels. This exceeds by an order of magnitude the largest dataset previously analyzed.

Finally, we present a theoretical framework and an algorithmic pipeline for capturing continuous conformational changes of the ribosome from ultralow-signal (-12dB) experimental cryo-EM. Our analysis shows a smooth, concerted change in molecular structure in two-dimensional projection, which might be indicative of the way the ribosome functions as a molecular machine.

The thesis ends with a summary and future prospects.

المنارة للاستشارات
www.manaraa.com

© Copyright by Ali Dashti, 2013
All Rights Reserved

www.manaraa.com

DEDICATION AND ACKNOWLEDGEMENTS

I would like to thank my supervisor, Dr. Roshan M. D'Souza. I am grateful and indebted to him for his sincere and valuable guidance and encouragement. In addition, I express my deepest appreciation and gratitude to my committee members and colleagues, Distinguished Professor Abbas Ourmazd and Dr. Peter Schwander, who ignited in me a long forgotten passion and perseverance for doing research at the cutting edge of science. I am thankful to my friends and colleagues Dr. Ahmad Hosseinzadeh and Ivan Komarov who contributed in the process of works presented in chapters two and three of this thesis. I would also like to thank Drs. Russell Fung and Dimitris Giannakis for their insightful comments and discussions regarding the work presented in Chapter four. I am grateful to my doctoral committee members, Professors Susan McRoy and Zeyun Yu for their valuable comments and discussions.

Last but not least, I am indebted to my parents, my father who thought me to always have an appetite for learning and my mother from whom I learnt to never settle in the pursuit of excellence. I dedicate this thesis to my wife, Reyhaneh, for without her early inspiration and enthusiasm, none of this could have happened.

المنارة للاستشارات
www.manaraa.com

TABLE OF CONTENTS

1. INTRODUCTION 2

1.1. SUMMARY 2

1.2. STRUCTURE AND CONFORMATIONAL SPECTRUM OF BIOLOGICAL MOLECULES 3

1.2.1. MOLECULAR MACHINES 3

1.2.2. STRUCTURAL VARIABILITY AND MOLECULAR FUNCTIONS 3

1.2.3. THE RIBOSOME 5

1.3. TECHNIQUES FOR DETERMINING STRUCTURE AND CONFORMATIONAL CHANGES 8

1.3.1. X-RAY CRYSTALLOGRAPHY 8

1.3.2. CRYOGENIC ELECTRON MICROSCOPY 10

1.3.3. X-RAY FREE ELECTRON LASER (XFEL) 11

1.4. EXTRACTING INFORMATION FROM LARGE DATASETS 13

1.4.1. STRUCTURE RECOVERY FROM CRYO-EM AND XFEL 13

1.4.2. MAPPING CONFORMATIONAL CHANGES BY CRYO-EM AND XFEL 16

1.5. IMPACT IN BIOLOGY AND MEDICINE 19

1.6. THESIS STATEMENT AND CONTRIBUTIONS 21

1.7. REFERENCES 25

2. A GPU ACCELERATED PARALLEL IMPLEMENTATION OF THE DIFFUSION MAP 29

2.1. SUMMARY 29

2.2. MANIFOLD EMBEDDING AND DIFFUSION MAP 30

www.manaraa.com

2.3. IMPLEMENTATION AND COMPUTATIONAL COSTS **33**

2.3.1. EXACT AND APPROXIMATE *K*-NN METHODS 34

2.3.2. HIGH-DIMENSIONAL DATA AND CURSE OF DIMENSIONALITY 36

2.3.3. BRUTE FORCE *K*-NN SOLUTIONS 37

2.3.4. IMPLEMENTATION ON GRAPHICAL PROCESSING UNITS 38

2.4. DISTRIBUTED PARALLEL PROCESSING ON GPU CLUSTER **40**

2.4.1. MASSAGE PASSING INTERFACE (MPI) 41

2.4.2. OPEN MULTIPROCESSING (OPENMP) 41

2.4.3. GRAPHICS PROCESSING UNITS (GPUS) 42

2.5. ALGORITHM DEVELOPMENT **44**

2.5.1. DISTRIBUTED *K*-NEAREST NEIGHBORS GRAPH CONSTRUCTION ON GPU CLUSTERS 45

2.5.2. BATCH INDEX SORTING 54

2.5.3. QUICK SELECT 55

2.6. PERFORMANCE BENCHMARKS **60**

2.6.1. BATCH INDEX SORTING 61

2.6.2. QUICK SELECT 66

2.7. CONCLUSIONS **69**

2.8. REFERENCES **70**

3. APPLICATIONS IN DETERMINING THE STRUCTURE AND CONFORMATIONS OF BIOLOGICAL ENTITIES **73**

3.1. SUMMARY **73**

المنارة للاستشارات

www.manaraa.com

3.2. HIGH RESOLUTION STRUCTURE OF VIRUSES FROM RANDOM DIFFRACTION SNAPSHOTS 74

3.2.1. BACKGROUND 74

3.2.2. METHODS 75

3.2.3. RESULTS 84

3.3. CONFORMATIONAL HETEROGENEITY IN MOLECULAR DEGREDATION 86

3.3.1. BACKGROUND 86

3.3.2. MODEL SYSTEM: MELTING ADK 93

3.3.3. RESULTS 94

3.4. CONCLUSIONS 99

3.5. REFERENCES 101

4. MAPPING CONFORMATIONAL SPECTRUM OF THE RIBOSOME 104

4.1. SUMMARY 104

4.2. BACKGROUND 105

4.3. METHODS 111

4.3.1. DATASET CHARACTERISTICS 111

4.3.2. PROJECTION DIRECTIONS 111

4.3.3. CONTRAST TRANSFER FUNCTION OF CRYOEM MICROGRAPHS 113

4.3.4. LOCAL EMBEDDING 115

4.3.5. NONLINEAR LAPLACIAN SPECTRAL ANALYSIS 117

4.3.6. CONNECTING PROJECTION DIRECTIONS 123

4.4. RESULTS 128

المنارة للاستشارات

www.manaraa.com

4.4.1. LOCAL EMBEDDINGS 128

4.4.2. OVERRULE SYSTEMATIC FACTORS 132

4.4.3. ISO-ORIENATIONAL MOVIE 137

4.4.4. ISO-CONFORMATIONAL MAP 139

4.5. CONCLUSIONS 140

4.6. REFERENCES 141

5. CONCLUSIONS AND FUTURE DIRECTION 143

5.1. CONCLUSIONS 143

5.2. DISCUSSIONS 144

5.3. FUTURE DIRECTIONS 147

6. CURRICULUM VITAE 150

المنارة للاستشارات

www.manaraa.com

LIST OF FIGURES

Figure 1: Bacterial translation cycle (Schmeing et al., 2009). 6

Figure 2: Ribosome structure and dynamics (Frank 2011). 7

Figure 3: Structure of T. thermophilus ribosome (Yusupov et al., 2001). 9

Figure 4: Schematic view of Cryo-EM imaging (Grassucci et al., 2007). 11

Figure 5: XFEL microscopy (Chapman et al., 2011). 12

Figure 6: Random conical tilt reconstruction (Leschziner, 2010). 14

Figure 7: Common line method of angular reconstruction (Van Heel 2010). 14

Figure 8: Maximum-likelihood unsupervised classification (Scheres 2010). 17

Figure 9: Schematic view of a manifold (Schwander et al., 2010). 31

Figure 10: Locality sensitive hashing (Haghani et al., 2008). 36

Figure 11: GPU Fermi architecture (Sanders and Kandrot, 2010). 43

Figure 12: CUDA Model (NVIDIA, 2008). 44

Figure 13 Data partitioning for distributed parallel execution. 48

Figure 14: Load balancing between cluster nodes 50

Figure 15: Computing k-NN in GPUs. 51

Figure 16: Task distribution within the nodes 53

Figure 17: Batch index sorting 55

Figure 18: Read in process. 57

Figure 19: Pivot process. 58

Figure 20: Write-out process. 59

Figure 21: Performance analysis of batch index sorting (varying k). 62

Figure 22: Performance analysis of batch index sorting (varying d). 64

Figure 23: Performance analysis of batch index sorting (varying n). 65

Figure 24: Quick-Select benchmarks against Garcia. 67

المنارة للاستشارات

www.manaraa.com

Figure 25: Quick-Select benchmarks against TBiS. 67

Figure 26: Timing benchmarks of quick select and MGPU select algorithms. 68

Figure 27: Comparison between Laplace-Beltrami operator spectra. 77

Figure 28: Association between pairs of (Ψ_i , Ψ_j) and Wigner D-functions. 78

Figure 29: Schematic diagram of the mirroring process. 79

Figure 30: Finding the mixing angle, θm. 80

Figure 31: Atomic-resolution structure of the satellite tobacco necrosis virus. 85

Figure 32: Comparison of the exact and recovered diffraction volumes. 86

Figure 33: Chignolin reconstruction from low radiation (Fung et al., 2008). 87

Figure 34: Detecting discrete conformations (Schwander et al., 2010). 90

Figure 35: Embedding of continuous conformations (Schwander et al., 2010). 91

Figure 36: ADK model system in 10 different conformation. 93

Figure 37: Diffusion map embedding of ADK. 97

Figure 38: ADK in closed conformation (Schwander et al., 2012). 99

Figure 39: Manifold of ribosome with two discrete conformations. 106

Figure 40: Diffusion map embedding results of ribosome cryo-EM snapshots. 108

Figure 41: Distribution of snapshots on orientation space (S2). 109

Figure 42: Schematic view of Lovisolo tessellations of a sphere. 112

Figure 43: Anisotropy of Cryo-EM dataset. 113

Figure 44: Contrast Transfer Function of a projection snapshot. 115

Figure 45 Raw (left) and average (right) 2D view of snapshots. 116

Figure 46: Local embedding gives one dimensional manifolds. 118

Figure 47: Geodesic distance as a measure of ordering along manifold. 119

Figure 48: Geodesic ordering. 120

Figure 49: Diffusion map embedding of concatenated data. 121

المنـــارة للاستشارات

www.manaraa.com

Figure 50: Topos (right) and chronos modes (left) of NLSA. **123**

Figure 51: Different steps of the iterative fitting process. **125**

Figure 52: Histogram of τ distribution over projection directions. **126**

Figure 53: Results of Histogram equalization. **127**

Figure 54: Local embedding of class 1. **128**

Figure 55: Local embedding of class 2. **129**

Figure 56: Local embedding of class 3. **129**

Figure 57: Distribution of three category of projection directions over S2. **130**

Figure 58: Re-embedding the tail manifold. **131**

Figure 59: Re-embedding the core. **132**

Figure 60: Correlation between direction order and defocus value. **133**

Figure 61: Correlation between direction order and difference defocus value. **134**

Figure 62: distribution of snapshots from tail of class 1 manifolds over S2. **134**

Figure 63: Correlation between direction order and angular distance over S2 **135**

Figure 64: Distribution pattern of tail manifold. **136**

Figure 65: Average from tail (a), core (b) and all (c) snapshots. **137**

Figure 66: NLSA Chronos modes. **138**

المنارة للاستشارات

www.manaraa.com

Chapter 1

Introduction

www.manaraa.com

1. Introduction

1.1. Summary

This chapter sets the stage for our contributions in the analysis of single-particle snapshots of biological entities. We start with a brief introduction to bimolecular structure and its relation to functional dynamics. We next present the ribosome molecule as an example of a molecular machine whose structural heterogeneity has an important role in its functional design. Single-particle imaging techniques such as Cryogenic Electron Microscopy (cryo-EM) imaging and X-ray Free Electron Laser (XFEL) diffraction for capturing the structure and its variations of biological entities are presented. A review of existing algorithmic tools for recovering high-resolution structure and conformational changes with each of the above techniques are discussed In addition, a section is dedicated to the impact of studying biological entities in high resolution and capturing structural variability using single-particle imaging techniques. We conclude the chapter with the thesis statement and objectives of this thesis is presented at the end of this chapter.

المنارة للاستشارات
www.manaraa.com

1.2. Structure and conformational spectrum of biological molecules

1.2.1. Molecular machines

The term "molecular machine" was introduced by Brue Albert's 1998 programmatic assay in the journal *Cell* (Alberts, 1998). "Machine" is a familiar term in macroscopic world. It brings the image of gears, levers and springs moving in concert with each other in a causal chain of events conducted mainly by gravity and inertia. However, the situation is fundamentally different in the aqueous environment of the cellular world, the traditional gravitational and inertia forces are dominated by the random thermal bombardment of the solvent molecules causing these nanometer-long "machines" to move in a jittery fashion. (Frank, 2011).

The main forces in this watery world are non-covalent interactions between molecular parts, levers and gears, and collision with water molecules. The most mysterious fact about molecular machines is their interactions and conformational changes of structure called functional dynamics that lead into groups of irreversible chemical reactions with directed motions towards making a specific product (Frank, 2011).

1.2.2. Structural variability and molecular functions

To perform their tasks, molecular machines need energy. The way the nano-scale molecules obtain their energy will reveal their statistic structural changes while working. This energy stems primarily from chemical reactions in the cell. A common chemical reaction for releasing energy is the hydrolysis of a high-energy phosphate compound, in the form of Adenosine triphosphate (ATP) or Guanosine triphosphate (GTP) in an

المنارة للاستشارات

www.manaraa.com

aqueous cell environment. In contrast to common view of machinery function, molecular machines do not have a smooth deterministic way of action and their functional motions are stochastic/random. The nano-scale structure and weak non covalent bonds between molecular subunits allow weak energies and thermal fluctuations form the cell environment to play a crucial role in this seemingly random behavior of biological molecules, best described by Brownian motor mechanisms of operations (Peskin et al., 1993; Cordova et al., 1992, Frank, 2011).

The nano-scale structure and weak non covalent bounds between molecular subunits allow weak energies and thermal fluctuations form the cell environment to. play a crucial role in this seemingly random behavior of biological molecules, best described by Brownian motor mechanisms of operations (Peskin et al., 1993; Cordova et al., 1992).

Brownian motor mechanisms of operations simply states that structural parts of molecular machines are subject to random stochastic changes lead by thermal fluctuations. However this random behavior in noisy environment cannot explain the *processivity* of molecular machines, the way molecules undergo unidirectional chemical reactions and produce their products. Directionality is explained by intervention of additional cofactors or binding effects that can lead to irreversible reaction steps and hence release of the product from molecular complex (Frank, 2011).

The Brownian motor mechanisms of operations is an important factor for interpretation of seemingly deterministic behavior of molecular machines from their captured "snapshots" at random time frames using single particle imaging. Functional states of molecular machines present the local minima in free energy landscape of molecule and

المنارة للاستشارات

www.manaraa.com

each molecule reaches different functional states by random motions as a result of thermal fluctuations.

1.2.3. The ribosome

The Ribosome is one of the most important molecular machines in nature. It is known as engineer of the cell. It plays a major role in protein synthesis, the central process for living cells. Ribosome is a large macromolecular machine, 250^{o} in diameter, with two distinct subunits mainly composed of ribosome RNA or rRNA in its core. Ribosome's main task is to knit the chain of polypeptides or infant proteins, from the sequences of amino acids delivered by transfer RNAs (tRNA) from genetic instructions encoded in messenger RNA (mRNA). This process known as translation is a central part of molecular biology dogma (Figure 1). Having such intricate and important tasks, the Ribosome is nearly preserved in all kingdoms of life.

Exploring Ribosome structure and its variability has been the subject of extensive research over the past decade (Schmeing et al., 2009). One target aim of these studies is to couple the conformational changes of different parts of ribosome with each other and provide a global view of conformational changes as a whole, which further provides an explanation of ribosome functional design in the process of translation.

Figure 2 demonstrates the overview of structure and dynamics of ribosome.The Ribosome structure is determined by the architecture of its two subunits. Ribosome has three tRNA binding sites, A (for amino-acyl), P (for peptidyl) and E (for exit) in the intersection of two subunits. The placement of tRNAs into these binding sites determines the stage of translation and consequent conformational changes (Frank 2011).

المنارة للاستشارات

www.manaraa.com

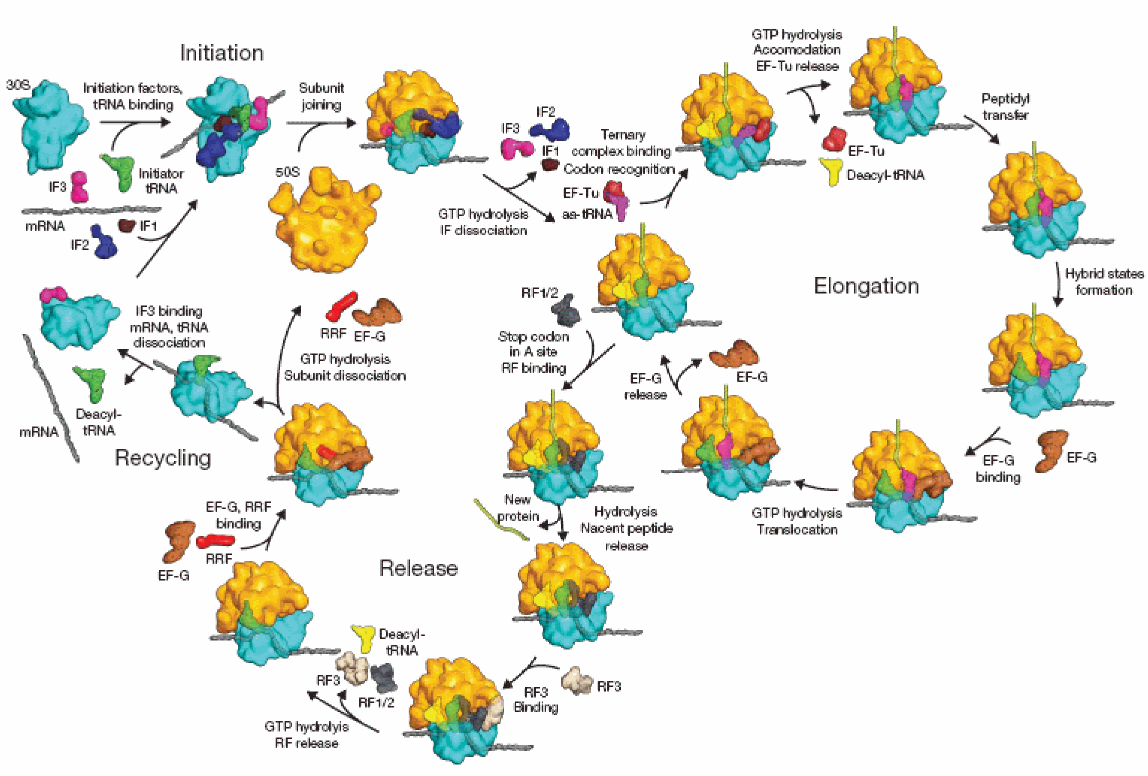


Figure 1: Bacterial translation cycle (Schmeing et al., 2009). The chain of reactions corresponding to initiation, elongation, release and recycling stages of translation are viewed schematically. Abbreviations are aa-tRNA for aminoacyl-tRNA; EF elongation factor; IF for initiation factor and RF for release factor.

As figure 2 demonstrates, ribosome subunits undergo various conformational changes. These degrees of freedom have evolved to perform the key actions of protein synthesis machinery. The question then arises is how to capture the structure and conformational changes of the Ribosome as a molecular machine during its functional states, and in addition, how to relate the structural variability of ribosome to its functional design.

www.manaraa.com

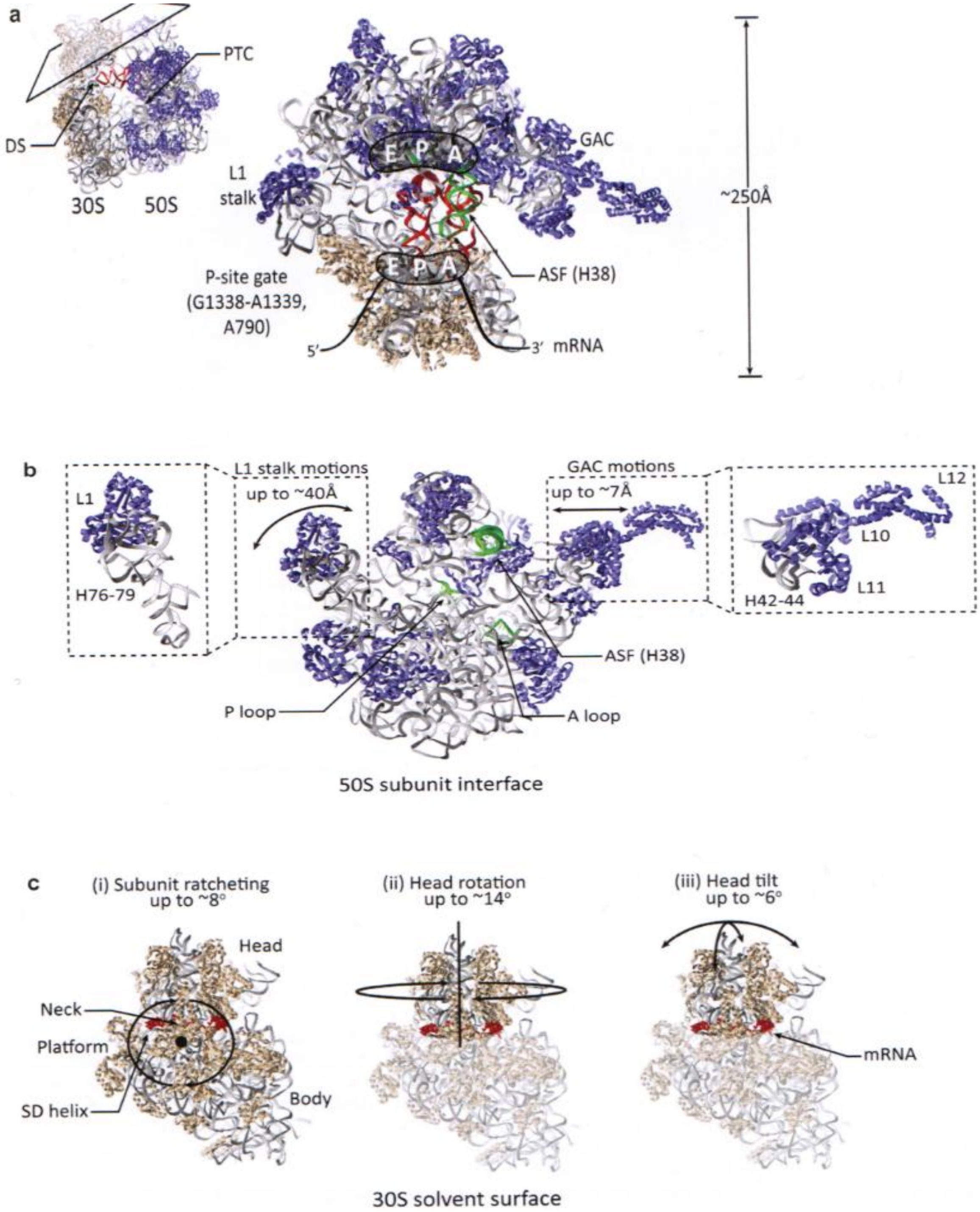


Figure 2: Ribosome structure and dynamics (Frank 2011). a) Ribosome and its subunits in protein translation machinery. b) Large subunit and estimated range of motion for L1 stalk and GAC components. c) small subunit and its range of motions.

www.manaraa.com

Our knowledge of Ribosome and its structural heterogeneity is hugely indebted to single particle imaging techniques and complex processing algorithms that enable us to capture and extract different conformational states of molecule while coexisting as sample ensemble all at once. This development will be augmented in future by further enhancement of experimental capabilities and development of more complex algorithms for time-resolved analysis. That would have a great influence on understanding the mechanism of translation and Ribosome machinery by revealing its tiny intermediate conformational changes. The new conformational spectrum will lead to discovery of novel functional states and provide a better view of ribosome functional dynamics (Fischer et al., 2010).

1.3. Techniques for determining structure and conformational changes

1.3.1. X-ray crystallography

In order to capture the conformational spectrum of a biomolecule, it is first necessary to know its' phenomenological description and structure in static state. X-ray crystallography is a well-established technique for this purpose. It is possible to capture high resolution structures of molecular machines with or without their functional ligands and even their subunits with X-ray crystallography. Reconstruction of bacterial ribosome structure is one of the major achievements of this technique (Yusupov et al., 2001) (Figure 3).

www.manaraa.com

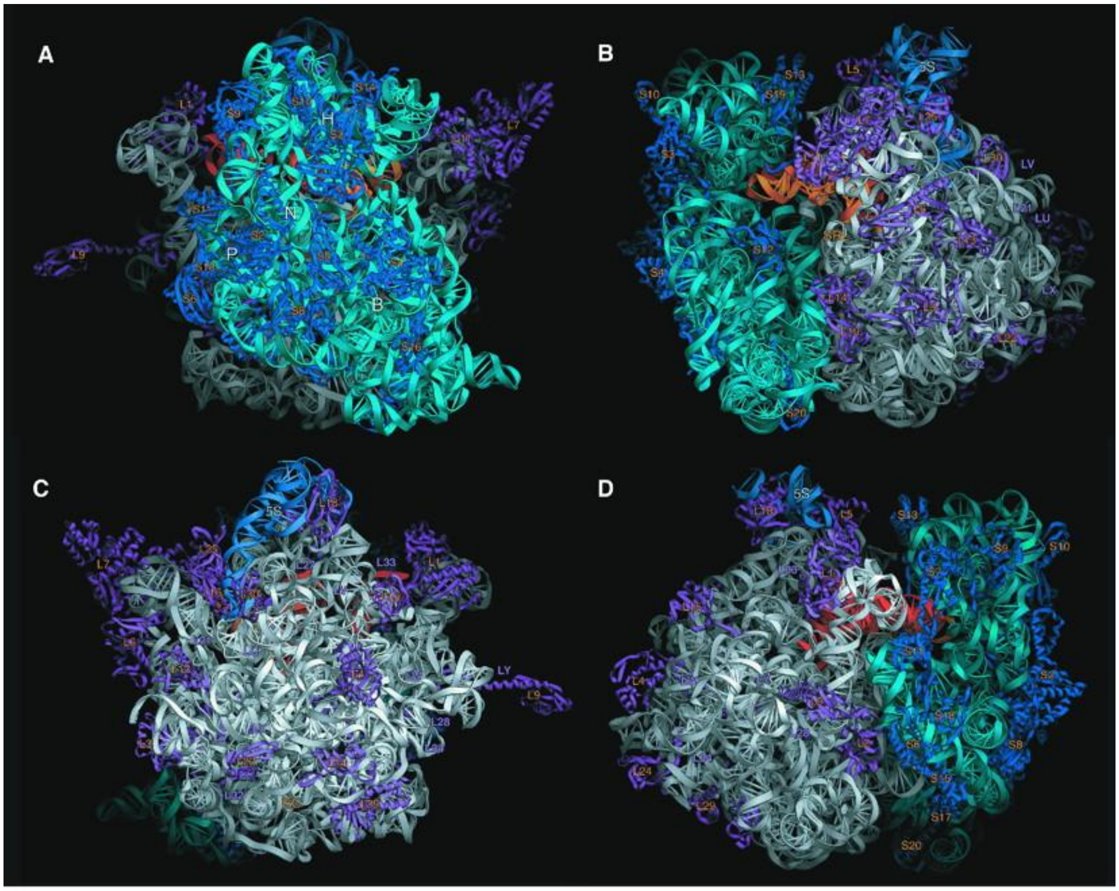


Figure 3: Structure of T. thermophilus ribosome (Yusupov et al., 2001). A) to D) represents 90o rotations of the molecule along vertical axis. Structure is determined with 5.5 Å resolution by X-ray crystallography.

X-ray crystallography determines atomic structure from crystals where the constituent atoms of biomolecule are placed in an ordered pattern. Crystals provide an structure view of "typical" molecule by averaging different conformations and leaving out study of conformational spectrum.

In order to step beyond mere structure reconstruction and observe the functional changes of a molecular machine, single particle imaging techniques such as cryogenic electron

المنارة للاستشارات

www.manaraa.com

microscopy (cryo-EM) and newly established X-ray free electron laser (XFEL) come into action.

1.3.2. Cryogenic electron microscopy

Cryogenic electron microscopy (cryo-EM) uses the same optical principles as the transmission electron microscopy (TEM) to provide two-dimensional (2D) projections of 3-dimensional (3D) biomolecule in a detector. Each 2D projection or *"projection snapshot"* is the line integral of the mean inner potential distribution representing the object in space. In order to provide a high vacuum path for electron beam, cryo-EM uses frozen-hydrated specimen preparation techniques (Taylor et al., 1976) in which multiple particles are embedded in a thin layer of ice before microscopy.

A cryo-EM micrograph represents a mosaic of projection snapshots, with each mosaic representing the snapshot of the object of interest in a random orientation and presumably in an instant of functional behavior. Figure 4 demonstrates a schematic view of cryo-EM and the resulting micrograph. For analyzing cryo-EM data, at first the micrograph is segmented into smaller boxes, each box containing one projection snapshot, by a process called particle picking (Rath and Frank 2004). The goal of image processing then will be to analyze projection snapshot of all particles to map conformational changes and recover structure of each conformation. As a results of such processing, a 3-D reconstruction of a molecular machine in each of its functional states is acquired from projection snapshots.

المنارة للاستشارات

www.manaraa.com

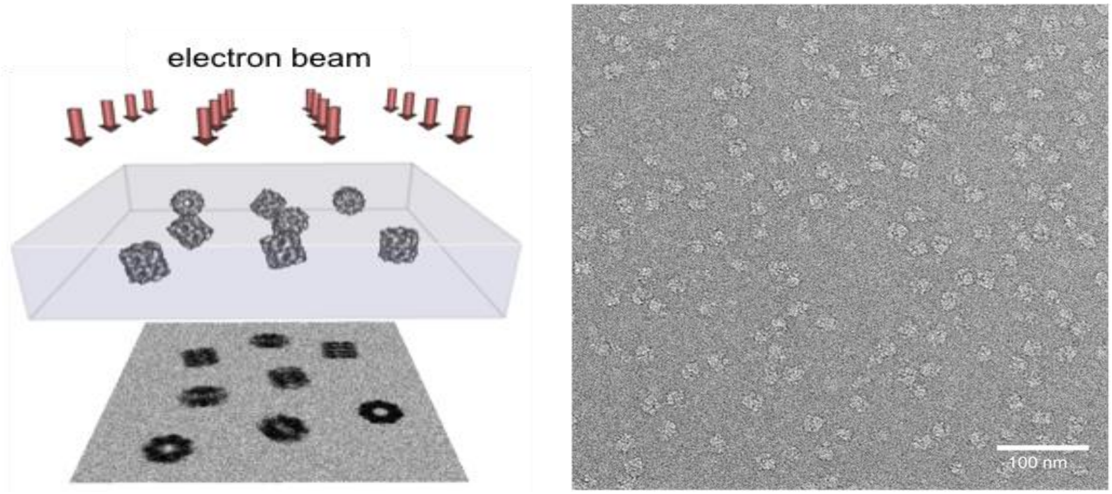


Figure 4: Schematic view of cryo-EM imaging (Grassucci et al., 2007). Distribution of projection snapshots in the micrograph reflects the particle distribution of ribosome molecules on a frozen-hydrated grid.

1.3.3. X-ray free electron laser (XFEL)

X-ray free electron laser have recently made it possible to have X-ray *diffraction snapshots* of biological entities before destroying the molecules by radiation damage (Boutet et al., 2012). X-ray strong ionizing effects almost make it impossible to have a diffraction pattern from single molecules. XFEL solves this issue by sending the ultra-short pulses and recording the diffraction snapshots before the disentanglement of molecules from intensity of the pulse (Gaffney and Chapman, 2007; Chapman et al., 2006). Radiation damage is a limiting factor in the quality of experimental structural data. In addition to noise, the resolution of spatial reconstruction of structure is determined by level of damages. The limitation is manifested in all single particle imaging techniques such as CryoEM (Frank 2002) and X-ray crystallography. XFEL ultra-short femto-second pulses that contains up to 10^{12} photons provide a possible solution to this problem

المنارة للاستشارات
www.manaraa.com

by obtaining diffraction snapshots of single particle beam before their destruction in a so called "scatter and destroy" approaches (Solem and Baldwin, 1982).

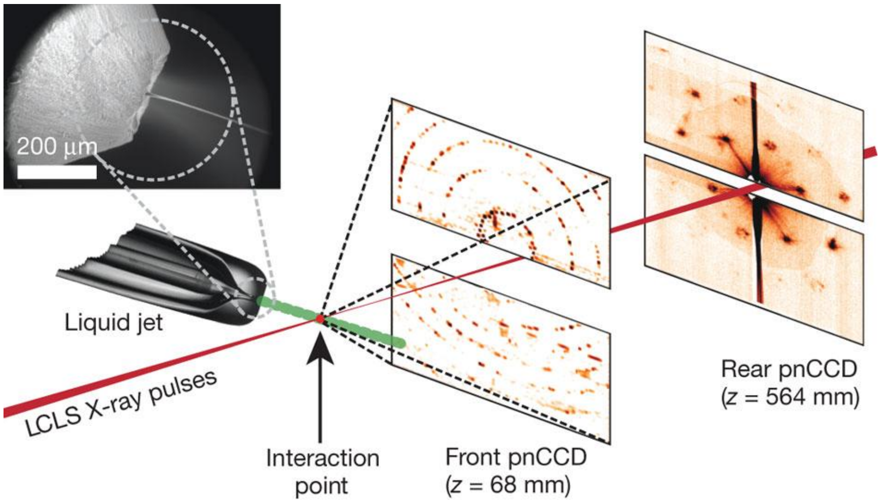


Figure 5: XFEL microscopy (Chapman et al., 2011). Nanocrystals of biological molecules injected from the liquid jet and have an interaction with X-ray FEL pulses in fornt of the detectors. Two detectors captures the high- and low-angle diffractions from X-ray FEL pulses.

Representing promising preliminary success in maintaining the natural state of molecules in incident beam (Hau-Reige et al., 2005) and recovering the scattering patterns before the start of radiation damage (Hau-Reige et al. 2007), XFEL is becoming a powerful tool for high resolution structure recovery and determination of conformational changes from single particle beams. Hence, there is an increasing need for noise robust algorithms to classify and reconstruct three-dimensional structures of functional states from XFEL diffraction patterns in the absence of orientation information and presence of overwhelming noise and background scattering.

المنارة للاستشارات
www.manaraa.com

1.4. Extracting information from large datasets

Both cryo-EM and XFEL generate large datasets of snapshots capturing 2D view of biological entities in random orientations. The presence of conformational heterogeneity in captured molecules adds an additional unknown factor for recovering structure from these techniques. In this section we review different methods of structure recovery and mapping conformational changes from XFEL and cryo-EM.

1.4.1. Structure recovery from cryo-EM and XFEL

In cryo-EM there is an active area for developing algorithms to determine the three-dimensional structure of single molecules. The main computational task in structure recovery however, is the determination of *a priory* unknown projection angles. In cryo-EM there is an established methodology with more than three decades worth of literature in this area that can be categorized in two class of algorithms for unfolding projection angles.

Method of common lines (Van Heel, 1987), finds the projection angles by having the assumption that all 2-D projections of a 3-D object have a common line in Fourier space. Having a common line between a set of three independent projection snapshots, relative Euler angles can be determined by finding associations between each subset of two snapshots (Figure 7).

Random-conical reconstruction, are aiming at constructing a conical projection geometry based on tilting the azimuthal in-plane position of the object and recover three Euler

المنارة للاستشارات

angles for each class of similar projection cones (Radermacher et al., 1987; Frank, 1998) (Figure 6).

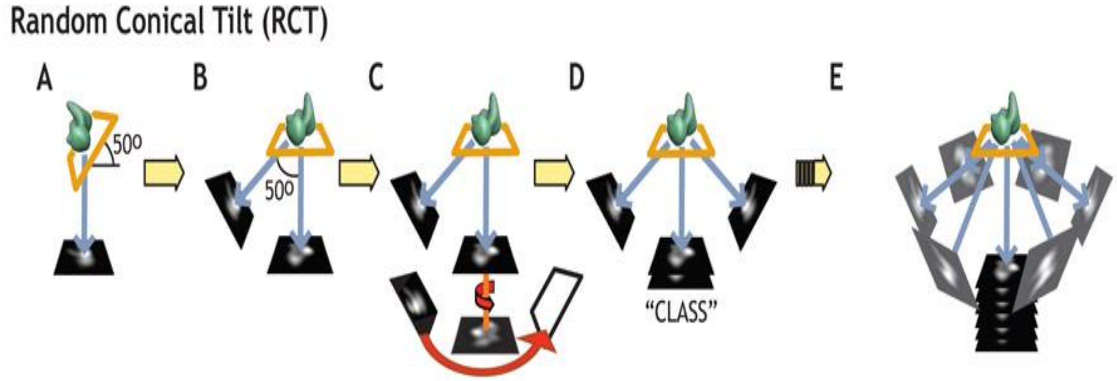


Figure 6: Random conical tilt reconstruction (Leschziner, 2010). A) Image is collected from tilted samples at 50^o. B) Second set of images are recorded from samples at 0^o. Two sets of images in A and B are aligned with each other by an in plane rotation. D) The two aligned images represents the same view of the molecule and can be viewed as they are in the same class. E) Having enough images to fully sample the desired structure, reconstruction can be made.

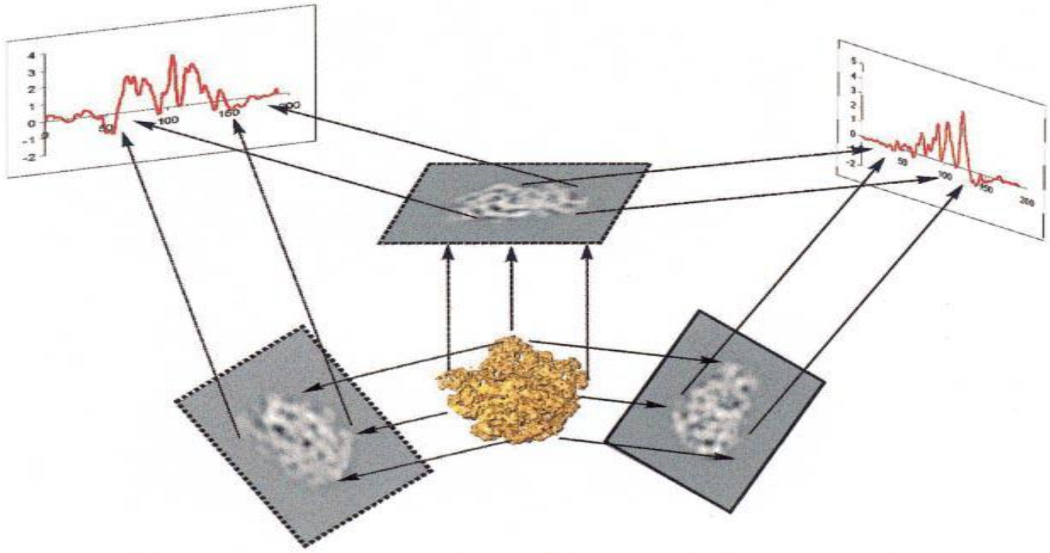


Figure 7: Common line method of angular reconstruction (Van Heel 2010). common line projection theorem stating that two different two-dimensional (2D) projections of the same 3D object have a one- dimensional (1D) line projection in common.

المنارة للاستشارات
www.manaraa.com

Our group presented the first demonstration of high resolution structure recovery from simulated XFEL snapshots of macromolecules using Generative Topographic Mapping (GTM) (Fung et al., 2008). The GTM approach is a Bayesian nonlinear factor-analytical approach that determines the structure using maximum-likelihood as a measure of classification for orientations. GTM fits the correlations between the snapshots in the presence of geometry constraints in an embedded space ('latent space') (Fung et al., 2008).

Other structure recovery solutions for XFEL (Elser, 2009), also associate snapshots to their orientation based on maximum likelihood classification. After finding the orientation angles, iterative phasing algorithms are then used to reconstruct a 3D diffraction volume by gathering information from all snapshots and their orientations and hence provide a 3D reconstruction of the molecule (Fineup, 1978; Oszlányi and Suto, 2004; Elser, 2003). However because of computational costs the best achievable reconstruction by Bayesian approaches has the resolution of 1/10th of molecule diameter (Moths and Ourmazd, 2011). At this resolution, in depth study of larger molecular entities and molecular machines such as viruses and large proteins are out of reach of these algorithms. There are non-Bayesian approaches with higher resolution of 1/30th of the object diameter (Giannakis et al., 2010 I; Schwander et al., 2012) but even this resolution is still not sufficient for studying viruses and large proteins such as ribosome (with > 200Å diameter) at atomic resolutions.

1.4.2. Mapping conformational changes by cryo-EM and XFEL

There are numerous algorithms for dealing with heterogeneity of large molecules in unknown orientations or projection angles from cryo-EM. Examples of such sophisticated algorithms in mapping conformational changes are the study of structural variability in 70S *E. coli* ribosome (Scheres et al., 2006), mapping structural heterogeneity of human fatty acid synthase (Brink et al., 2004) and tRNA movement and its role in translation dynamics by time-resolved electron microscopy (Fischer et al., 2010). Despite the collective evidence for crucial role of conformational heterogeneity in cellular life, the study of structural variability is still in its early stages having limited abilities for finding structure and dynamics and relating them into function.

The first set of algorithms for mapping conformational changes in cryo-EM includes supervised classification methods which groups the projections based on their similarity to few *a-priori* known 2D or 3D molecular models (Valle et al., 2002). These set of algorithms have two limiting assumptions. First, they assume a known atomic model for all functional states of a molecule. This assumption might not be valid in some molecular machines with unknown functional states and novel structures. Secondly, and similar to all classification algorithms, these methods have an *a-priori* known number classes for functional states of a molecule. This limitation forces the algorithm to detect the desired number of classes and gives no information about the actual number of functional states that are found in the real world

There are also unsupervised methods of classification that requires little or no *a-priori* knowledge about the classes. Methods such as maximum-likelihood classification

المنارة للاستشارات

www.manaraa.com

(Scheres, 2010) (Figure 8) and bootstrapping (Liao and Frank 2010) perform the classification completely unsupervised. Unsupervised algorithms do not have the first limitation of supervised classification approaches. However the second limitation still holds. By choosing a cluster number for conformations, these algorithms are limiting the functional states of molecules to the assigned number.

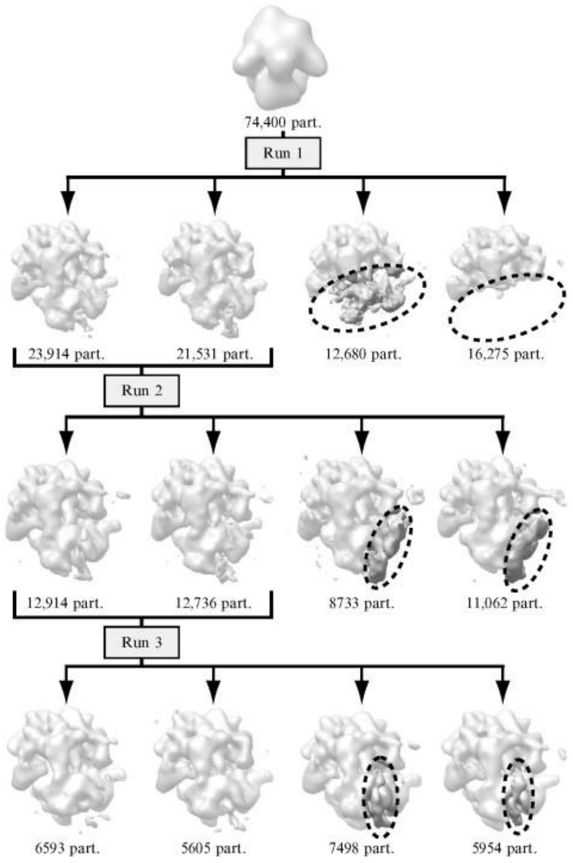


Figure 8: Maximum-likelihood unsupervised classification (Scheres 2010). Projection snapshots of ribosome is processed with ML3D software. Three iterative runs of the algorithm separates the molecules from incomplete subunits (first run), empty ribosomes from the one with strong t-RNA signal (second run) and weak t-RNA signals (Third run).

Structural heterogeneity of ribosome has been the subject of extensive research in Cryo-EM in recent years. Due to its important role in the process of cellular translation, capturing translating ribosome in motion has many applications in understanding

www.manaraa.com

functional units and dynamics of translation in general. However most of the computational algorithms for mapping conformational changes from cryo-EM data stem from classification roots (Scheres et al., 2006;2010). A typical result of cryo-EM processing algorithm reveals functional dynamics of ribosome in different phases of translation as discrete functional states, having a presupposed number of classes. As it is discussed in above, a priory given number of classes may not entail any biological and physical significance and in result, two or more of functional states may indeed have the same structure and be merged manually after the algorithms' run.

In a recent work by (Fischer et al., 2010), the authors suggested a time-resolved detection of continuous conformational changes by extracting some loose timing information from experimental cryo-EM setup. As a result, 50 distinct conformational states of ribosome during translocation, a phase in translation process (Figure 1), is detected with unbiased computational sorting using loose timing. Three-dimensional reconstruction of the conformations showed many intermediate states for tRNAs which led to better understanding of tRNA movement through the ribosome. In addition, functional dynamics of ribosome is revealed along the motion of tRNA in both head and body of its subunits. It is believed that these conformational changes are helping the tRNA to position along the neck of ribosome subunits.

For having loose timing information, cryo-EM samples were prepared at different time points (zero, one two, five and twenty minutes) after adding the tRNAs to a solvent with empty ribosomes. Zero point in time is assumed to be the starting point of reaction and by

المنارة للاستشارات

sampling ribosome in different time frames, projection snapshots entail a loose time stamp which allocate ribosome in different phases of its machinery.

There are proposed algorithms for mapping conformational changes (Schwander et al., 2010) and sorting ensembles of particle mixtures (Tegze and Bortel 2013) from simulated X-ray diffraction patterns. In addition, the other pathway of capturing conformational heterogeneity form XFEL is to exploit timing information from the experimental setting providing different states of a biomolecule based on pre-known phases of a reaction or molecular function leading to 3D reconstruction of non-identical objects with limited resolution (Schmidt et al., 2008; Bergh et al., 2008).

However, there are as yet no experimental results, capturing conformational changes from XFEL diffraction patterns. This is mainly because of the fact that XFEL experimental setup is still in its infancy and have many experimental and algorithmic challenges for structure recovery of macromolecular ensembles yet aside conformational changes.

1.5. Impact in biology and medicine

Molecular machines, the basic functional units of the cell almost and always undergo conformational changes in different degrees due to free energy landscape constrained by their structure. Ideally, .single-particle approaches are perfect candidate for determining structural variability of biological entities. There is an increasing appreciation for the fact that the dynamic behavior of molecular machines yields significant information about

المنارة للاستشارات
www.manaraa.com

their functional design. High-resolution determination of structural variability is an important task mainly for its role in scientific understanding of cellular functions in their basic level by studying the building blocks of the cell such as proteins, molecular agents and in higher levels chromosomes and organelles.

A deep understanding of the mechanisms and role of conformational variability in biological entities would revolutionize our knowledge of cellular life and its key processes in both normal and pathological states (Schwander et al., 2010). Examples of importance of structural variability in clinical setting are abundant and range from reversible, pH-driven conformational changes of flavivirus and their role in the host stabilization and processing response (King and Kesson, 2003) to the classical descriptions hemoglobin R to T switch and its role in oxygen transportation in blood cells (Schotte et al., 2003) in addition to the relation between virulence of the dengue virus to structural rearrangements in protein binding sites (Yu et al., 2008). Unraveling this relation is expected to help discovery of novel strategies for fighting viral infection (Schwander et al., 2010).

The impact of studying structure and dynamics of ribosome in the fields of medicine and biology can be deduced from the fact that it awarded two Nobel prizes in 1974, in Physiology or Medicine for its discovery and 2009, in Chemistry for detailed structure and mechanisms.

Quoting from press release of 2009 Chemistry Nobel prize:

"An understanding of the ribosome's innermost workings is important for a scientific understanding of life. This knowledge can be put to a practical and immediate use; many

المنارة للاستشارات
www.manaraa.com

of today's antibiotics cure various diseases by blocking the function of bacterial ribosome. Without functional ribosome, bacteria cannot survive. This is why ribosome is such an important target for new antibiotics."

Hereby, the understanding of ribosome and its dynamics in fine levels of resolution, one of the main perspectives of this thesis, would play a crucial role in future medicine and molecular biology.

1.6. Thesis statement and contributions

This thesis aims to study the feasibility of applying manifold embedding-based algorithms for structure recovery, and mapping conformational changes of biological entities from large datasets of single particle imaging snapshots. Towards this goal, I have developed a scalable and highly efficient implementation of Diffusion Maps (Coifman and Lafon 2006), a non-linear non-Bayesian approach for dimensionality reduction. The suite of software implementations will be used in our groups' endeavors in structure recovery in the presence of structural variability (Schwander et al., 2010; Giannakis et al., 2010).

This project has four specific objectives:

First: To develop an optimized, fast and expandable algorithm for Diffusion Map embedding and its' most computationally expensive part, *k*-NNG, using a computational cluster of graphical processing units.

المنارة للاستشارات

www.manaraa.com

Second: To apply the developed Diffusion Map algorithm for three-dimensional reconstruction of Satellite Tobacco Necrosis virus (STNV) from simulated random diffraction snapshots up to atomic resolutions, the highest resolution attainable by any reconstruction method.

Third: To apply the developed Diffusion Map algorithm for unsupervised detection of conformational changes of Adelynate Kinase (ADK) from simulated random diffraction snapshots, for up to ten conformational states, the highest number of conformational states attainable by any method. This objective provides further proof for extendibility of implementation by moving number of snapshots up to an order of magnitude higher than the upper boundaries of the present XFEL and CryoEM setups.

Forth: To design a theoretical pipeline using developed algorithms and gained knowledge of previous steps in order to map conformational changes of the Ribosome macro-molecule from random snapshots of experimental cryo-EM.

My contributions towards the aim of this thesis are as follows:

I designed (with help of RMD[1] and IK[2]) and developed a fast and scalable implementation of the Diffusion Map algorithm capable of leveraging the parallel computing capabilities of a cluster of compute nodes with graphics processing units (GPUs).

---

[1] Roshan M. D'Souza
[2] Ivan Komarov

www.manaraa.com

The said implementation was applied successfully in a software pipeline for structure recovery (by AH[3] and PS[4]) from large dataset of Satellite Tobacco Necrosis Virus (STNV) simulated XFEL diffraction snapshots.

I (with help of PS) applied the implementation for mapping discrete conformational changes of melting Adenylate Kinase (ADK) molecule from a large simulated dataset of XFEL snapshots.

In addition, I designed (with help of AO[5] and PS) and developed an algorithmic pipeline, including Diffusion Map, for mapping the continuous conformational changes of ribosome from experimental cryo-EM dataset.

The rest of the thesis is organized as follows: In chapter two, Diffusion Map, a manifold-based method for nonlinear dimensionality reduction is explained, and algorithm development steps for having a fast and scalable implementation of Diffusion Map on a GPU cluster is described. Chapter three presents the results of applying Diffusion Map to two model systems aiming at structure recovery and mapping conformational changes. At first, theoretical framework and algorithm development for high-resolution structure recovery of STNV is described and then we show the results of Diffusion Map embedding for detecting 10 discrete conformational changes of ADK molecule undergoing the melting process. Chapter four presents the theoretical framework and algorithmic pipeline developed and its application to mapping structural heterogeneity of

---

[3] Ahmad Hosseinizadeh
[4] Peter Schwander
[5] Abbas Ourmazd

www.manaraa.com

the ribosome from experimental cryo-EM. The final chapter concludes the thesis with a discussion of the remaining challenges and a road map for future work.

www.manaraa.com

1.7. References

Alberts, Bruce. "The cell as a collection of protein machines: preparing the next generation of molecular biologists." *Cell* 92.3 (1998): 291-294.

Bergh, Magnus, et al. "Feasibility of imaging living cells at subnanometer resolutions by ultrafast X-ray diffraction." *Q. Rev. Biophys* 41 (2008): 181-204.

Boutet, Sébastien, et al. "High-resolution protein structure determination by serial femtosecond crystallography." *Science* 337.6092 (2012): 362-364.

Brink, Jacob, et al. "Experimental verification of conformational variation of human fatty acid synthase as predicted by normal mode analysis." *Structure* 12.2 (2004): 185-191.

Chapman, Henry N., et al. "Femtosecond diffractive imaging with a soft-X-ray free-electron laser." *Nature Physics* 2.12 (2006): 839-843.

Chapman, Henry N., et al. "Femtosecond X-ray protein nanocrystallography." *Nature* 470.7332 (2011): 73-77.

Coifman, Ronald R., and Stéphane Lafon. "Diffusion Maps." *Applied and computational harmonic analysis* 21.1 (2006): 5-30.

Cordova, Nicolas J., Bard Ermentrout, and George F. Oster. "Dynamics of single-motor molecules: the thermal ratchet model." *Proceedings of the National Academy of Sciences* 89.1 (1992): 339-343.

Elser, Veit. "Phase retrieval by iterated projections." *JOSA A* 20.1 (2003): 40-55.

Elser, Veit. "Reconstruction algorithm for single-particle diffraction imaging experiments." *Physical Review E* 80.2 (2009): 026705.

Fienup, James R. "Reconstruction of an object from the modulus of its Fourier transform." *Optics letters* 3.1 (1978): 27-29.

Fischer, Niels, et al. "Ribosome dynamics and tRNA movement by time-resolved electron cryomicroscopy." *Nature* 466.7304 (2010): 329-333.

Frank, Joachim. "How the Ribosome Works: For over 40 years, scientists have studied the machine that synthesizes proteins. Yet, until recently, no one had" seen" the ribosome with any great clarity." *American scientist* 86.5 (1998): 428-439.

Frank, Joachim. "Single-particle imaging of macromolecules by cryo-electron microscopy." *Annual review of biophysics and biomolecular structure* 31.1 (2002): 303-319.

Frank, Joachim. *Molecular machines in biology*. 2011.

Fung, Russell, et al. "Structure from fleeting illumination of faint spinning objects in flight." *Nature Physics* 5.1 (2008): 64-67.

Gaffney, K. J., and H. N. Chapman. "Imaging atomic structure and dynamics with ultrafast X-ray scattering." *Science* 316.5830 (2007): 1444-1448.

Giannakis, Dimitrios, Peter Schwander, and Abbas Ourmazd. "The symmetries of image formation by scattering. I. Theoretical framework." *arXiv preprint arXiv:1009.5035* (2010).

المنارة للاستشارات
www.manaraa.com

Grassucci, Robert A., Derek J. Taylor, and Joachim Frank. "Preparation of macromolecular complexes for cryo-electron microscopy." *Nature protocols* 2.12 (2007): 3239-3246

Hau-Riege, Stefan P., et al. "Pulse requirements for x-ray diffraction imaging of single biological molecules." *PHYSICAL REVIEW-SERIES E-* 71.6 (2005): 061919.

Hau-Riege, Stefan P., et al. "Subnanometer-scale measurements of the interaction of ultrafast soft x-ray free-electron-laser pulses with matter." *Physical review letters* 98.14 (2007): 145502.

Hosseinizadeh, A., et al. "High-Resolution Structure of Viruses from Random Snapshots." *arXiv preprint arXiv:1303.7253* (2013).

King, Nicholas JC, and Alison M. Kesson. "Interaction of flaviviruses with cells of the vertebrate host and decoy of the immune response." *Immunology and cell biology* 81.3 (2003): 207-216.

Leschziner, Andres. "Chapter Nine-The Orthogonal Tilt Reconstruction Method." *Methods in enzymology* 482 (2010): 237-262.

Liao, Hstau Y., and Joachim Frank. "Classification by bootstrapping in single particle methods." *Biomedical Imaging: From Nano to Macro, 2010 IEEE International Symposium on*. IEEE, 2010.

Moths, Brian, and Abbas Ourmazd. "Bayesian algorithms for recovering structure from single-particle diffraction snapshots of unknown orientation: a comparison." *Acta Crystallographica Section A: Foundations of Crystallography* 67.5 (2011): 481-486.

Oszlányi, Gábor, and Andras Suto. "Ab initio structure solution by charge flipping." *Acta Crystallographica Section A: Foundations of Crystallography* 60.2 (2004): 134-141.

Peskin, Charles S., Garrett M. Odell, and George F. Oster. "Cellular motions and thermal fluctuations: the Brownian ratchet." *Biophysical Journal* 65.1 (1993): 316-324.

Radermacher, M., et al. "Three-dimensional reconstruction from a single-exposure, random conical tilt series applied to the 50S ribosomal subunit of Escherichia coli." *Journal of microscopy* 146.2 (1987): 113-136.

Rath, B. K., and J. Frank. "Fast automatic particle picking from cryo-electron micrographs using a locally normalized cross-correlation function: a case study." *Journal of structural biology* 145.1 (2004): 84-90.

Scheres, Sjors HW, et al. "Disentangling conformational states of macromolecules in 3D-EM through likelihood optimization." *Nature Methods* 4.1 (2006): 27-29.

Scheres, Sjors HW. "Chapter Eleven-Classification of Structural Heterogeneity by Maximum-Likelihood Methods." *Methods in enzymology* 482 (2010): 295-320.

Schmeing, T. Martin, and V. Ramakrishnan. "What recent ribosome structures have revealed about the mechanism of translation." *Nature* 461.7268 (2009): 1234-1242.

Schmidt, K. E., et al. "Tomographic femtosecond X-ray diffractive imaging." *Physical review letters* 101.11 (2008): 115507.

Schotte, Friedrich, et al. "Watching a protein as it functions with 150-ps time-resolved X-ray crystallography." *Science* 300.5627 (2003): 1944-1947.

المنارة للاستشارات

www.manaraa.com

Schwander, Peter., et al. "Mapping the conformations of biological assemblies." *New Journal of Physics* 12.3 (2010): 035007.

Schwander, Peter, et al. "The symmetries of image formation by scattering. II. Applications." *Optics Express* 20.12 (2012): 12827-12849.

Solem, Johndale C., and George C. Baldwin. "Microholography of living organisms." *Science* 218.4569 (1982): 229-235.

Taylor, Kenneth A., and Robert M. Glaeser. "Electron microscopy of frozen hydrated biological specimens." *Journal of ultrastructure research* 55.3 (1976): 448-456.

Tegze, Miklós, and Gábor Bortel. "Selection and orientation of different particles in single particle imaging." *Journal of structural biology* 183.3 (2013): 389-393.

Valle, Mikel, et al. "Cryo-EM reveals an active role for aminoacyl-tRNA in the accommodation process." *The EMBO journal* 21.13 (2002): 3557-3567.

Van Heel, Marin. "Angular reconstitution: a posteriori assignment of projection directions for 3D reconstruction." *Ultramicroscopy* 21.2 (1987): 111-123.

Van Heel, Marin, et al. "Single-particle electron cryo-microscopy: towards atomic resolution." *Quarterly reviews of biophysics* 33.4 (2000): 307-369.

Yu, I-Mei, et al. "Structure of the immature dengue virus at low pH primes proteolytic maturation." *Science* 319.5871 (2008): 1834-1837.

Yusupov, Marat M., et al. "Crystal structure of the ribosome at 5.5 Å resolution." *science* 292.5518 (2001): 883-896.

المنارة للاستشارات

Chapter 2

A GPU accelerated parallel implementation of the Diffusion Map

www.manaraa.com

2. A GPU accelerated parallel implementation of the Diffusion Map

2.1. Summary

In this chapter, we first provide a brief introduction to diffusion map, a nonlinear dimensionality reduction algorithm. We next describe a GPU accelerated, computing cluster-based parallel implementation of this algorithm. We describe efficient data partitioning and communication schemes that were developed to partition the large input data set among individual nodes of a computing cluster for balanced execution. The k-Nearest Neighbor Graph (k-NNG) construction, a step in the Diffusion Map algorithm is a two step process. The first step finds distances between individual images based on a pre-defined metric. The second finds the k-Nearest Neighbors based on this metric. We have developed an efficient implementation for computing distances using matrix multiplication formulation for which there exist optimized libraries. For the second step we describe two novel based on multi-sort and partial sort. Overall, our work delivered a 13x speedup over a comparable parallel implementation on a CPU cluster.

www.manaraa.com

2.2. Manifold embedding and Diffusion Map

Manifold embedding exploits the fact that the similarities between high-dimensional data points reveal themselves as a low-dimensional hyper-surface embedded in the high-dimensional data space. The embedded hyper-surface, called a manifold from here on, contains the information about the object giving rise to the snapshots, for example, 3D structure of a virus or conformational states of a molecular machine.

Manifold embedding measures the similarities or nonlinear correlations between clouds of snapshots by tracking their response to an operator and finding the orthonormal coordinates needed to describe the manifold. Another interesting feature of manifolds is their dimensionality. Dimensionality of a manifold has a direct relationship with the degrees of freedom the object. For example, since rotation of an object in space has three degrees of freedom and can be characterized by three values (Euler angles), the manifold of snapshots form a three-dimensional hyperspace in the nine-dimensional space of rotation (SO(3)). (Figure 9).

Manifold embedding is distinguished from graph-theoretic approaches of dimensionality reduction due to its presupposition of finding manifold in low-dimensional space that lead to exploiting the intrinsic geometry of the data. In contrast, graph-theoretic approaches are more general and have no specific "geometric" treatment of data.

المنارة للاستشارات

www.manaraa.com

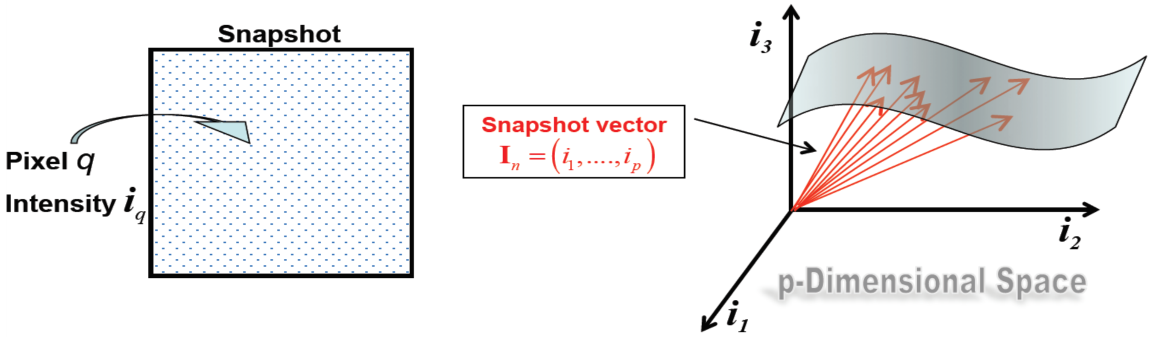


Figure 9: Schematic view of a manifold (Schwander et al., 2010). Correlations in high-dimensional space reveal themselves as a low-dimensional hyper-surface (manifold).

Diffusion Map is a manifold-based approach that uses the so-called diffusion characteristics of the dataset. Specifically, it measures the distance between two points according to their Euclidean distance, but the path taken by the diffusion of heat. An affinity graph is defined as the exponential normalization of Euclidean distance between points with their neighbors in the *k*-nearest neighbor graph. Probability of diffusion from one point to another on a path in the manifold is then defined by applying Laplace-Beltrami normalization to the affinity graph. A diffusion probability matrix contains all the probabilities of diffusion from one point to another on the manifold. Finally, eigenvectors of the diffusion probability matrix provide a description of the manifold in terms of a set of Euclidean coordinates (eigenvectors).

A flowchart of a Diffusion Map algorithm is demonstrated in the following:

www.manaraa.com

Diffusion Map embedding, following (Coifman and Lafon, 2006)
Inputs:

n×d snapshots' pixel intensity matrix: I
Gaussian width: ε
Normalization parameter: α
nearest neighbors parameter: k
embedding space dimension: t
Output:

Diffusion Map Ψ: $\mathbb{R}^d \rightarrow \mathbb{R}^t$

Steps:

1: Construct n×k *k*-nearest neighbor distance S and index N matrices, from Euclidean distances between rows of I.

2: construct an n×n sparse symmetric weight matrix W, such that:

$$W_{ij} = \begin{cases} 1, & if\ i = j \\ e^{-S_{ij}^2}, & if\ j = N_{ik} \\ W_{ji}, & if\ W_{ij} \neq 0 \\ 0, & otherwise. \end{cases}$$

3: Calculate the n×n diagonal matrix Q with nonzero elements $Q_{ii} = \sum_{j=1}^{s} W_{ij}$

4: Form the anisotropic kernel matrix $K = Q^{-\alpha} W Q^{-\alpha}$

5: Calculate the n×n diagonal matrix D with nonzero elements $D_{ii} = \sum_{j=1}^{s} K_{ij}$

6: Form n×n sparse transition probability matrix $P_\varepsilon = D^{-1} K$

7: Return Diffusion Map Ψ, as a result of standard Eigen functions decomposition problem $P_\varepsilon \nu = \lambda \nu$, such that:

$$\Psi \triangleq \begin{cases} \lambda_1 \nu_1 \\ \lambda_2 \nu_2 \\ \dots \\ \lambda_t \nu_t \end{cases}$$

المنارة للاستشارات

www.manaraa.com

2.3. implementation and computational costs

The main computational part of Diffusion Map is construction of a *k*-NN distance graph for a high-dimensional data cloud that will be analyzed to generate a Diffusion Map in subsequent stages of the algorithm. The input data to the manifold embedding application contains upwards of 10^7 images, with each data point having up to 160k pixels.

The *k*-Nearest Neighbor (*k*-NN) and the related *k*-Nearest Neighbor Graph (*k*-NNG) are important techniques used in a variety of fields such as pattern recognition (Cover and Hart, 1967), genomics (Dudoit et al., 2002), data mining (Levin, 2004), navigation systems (Safar, 2005) and fluids dynamics (Adams et al., 2007). The *k*-NN search is rooted in the post office problem first mentioned by Donald Knuth in The Art of Computer Programming (Knuth, 2006). Post-office problem is the task of finding nearest post office(s) for each resident from a set of post offices in the area.

Given a set of vectors $\{x_1, x_2, \ldots, x_n\} \in \mathbb{R}^D$, a distance metric $D: \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}$, nearest neighbor number k and query vector $q \in \mathbb{R}^D$

The nearest neighbor search set is the following:

$$\{x_{q1}, x_{q2}, \ldots, x_{qk}\} \text{ where } \{q_1, q_2, \ldots, q_k\} \subset \{1 \ldots n\} \text{ and}$$

$$D(q, x_{q1}) \leq D(q, x_{q2}) \leq \ldots \leq D(q, x_{qk})$$

and

$$D(q, x_{qk}) \leq D(q, x_i) \text{ for } all\ i \in \{1 \ldots n\} - \{q_1, q_2, \ldots, q_k\}$$

As opposed to *k*-NN search, in *k*-NNG construction, every vector in the reference

www.manaraa.com

data set is also a query vector. In the following, nearest neighbor set, for each x_i that belongs to reference set ($i \in \{1,2,...n\}$), is shown:

$$\{x_{i1}, x_{i2},...,x_{ik}\} \text{ where } \{i_1, i_2,...,i_k\} \subset \{1...n\} \text{ and}$$

$$D(x_i, x_{i1}) \leq D(x_i, x_{i2}) \leq ... \leq D(x_i, x_{ik})$$

and

$$D(x_i, x_{ik}) \leq D(x_i, x_j) \text{ for } \textit{all } j \in \{1...n\} - \{i_1, i_2,...,i_n\}$$

In this section, we discuss previous work on algorithms for *k*-nearest neighbor graph construction and *k*-nearest neighbor search. Broadly speaking, there are two categories of algorithms. One set of algorithms provides the exact *k*-NNs while the second set of algorithms sacrifices accuracy to a certain degree to gain on performance. For low-dimensional data sets, there are several efficient algorithms based on space partitioning data structures. For intermediate dimensional data sets, there are approximate algorithms based on hashing (Datar et al., 2004). For high-dimensional data sets where exact computation of *k*-NN is a must, the only alternative is a brute force search that is computationally quite expensive. Recently, there has been much research into addressing the computational complexity of a brute force search by using novel architectures such as graphics processing units.

2.3.1. Exact and approximate *k*-NN methods

In very low-dimensional spaces (2D or 3D) graph-based searching methods such as those using Voronoi diagrams and proximity graphs are very efficient and achieve super-linear

المنارة للاستشارات
www.manaraa.com

speedup. For higher dimensions, space partitioning methods such as kd-Trees (Bentley, 1990) , BBD-Trees (Arya et al., 1994), rp-Trees (Dasgupta and Freund, 2008) and Metric-Trees are most efficient. For these methods, there is a pre-processing step where the reference data set is used to build the search data structure. Once this data structure is built, the search for *k*-NN is quite cheap. However, building the search data structure itself is quite expensive . Therefore these methods are most useful if the reference data set is static. For large data dimensions and large values of k, the search process is no better than the brute-force technique.

In many applications, especially with large data dimensions, search accuracy constraints are not very stringent. In such cases, methods have been developed based on hashing to bin objects based on the object hash that is a function of its vector coordinates. Popular hashing schemes include the Locality Sensitive Hashing (LSH) (Datar et al., 2004), Z-Morton Curve based hashing (Connor and Kumar, 2010) and Hilbert space filling curve hashing (Moon et al., 2001) . Essentially, hashing schemes convert a *D*-dimensional problem to a 1-D problem through a hash function that preserves proximity of the input data set (Figure 10). The most challenging part of this scheme is the definition of a good hashing function. There are other approximate algorithms that use a hybrid approach. For example, in (Connor and Kumar, 2010), a disk-based quad-tree data structure is initially used to partition the reference data set. The *k*-NN search is conducted in two phases. A Z-order-based approximate proximity measure is used to find the approximate *k*-NN. Next, a recursive correction algorithm is used to improve the accuracy.

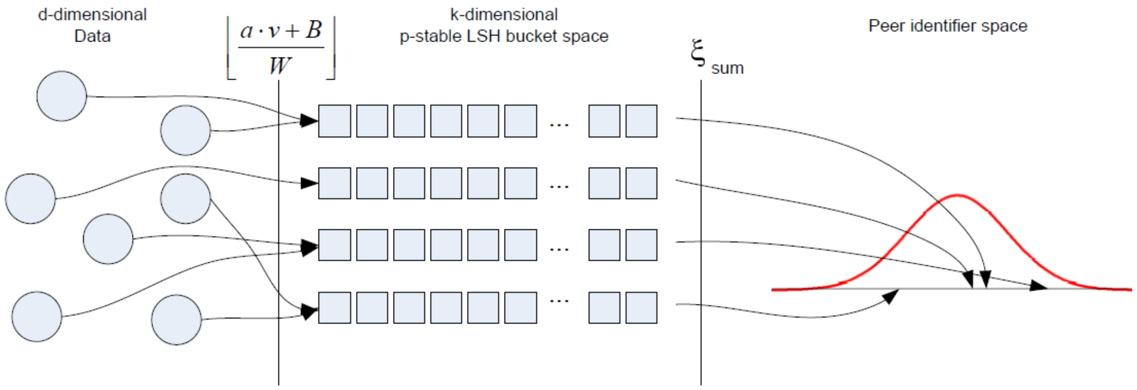

d-dimensional
Data
$\left\lfloor \frac{a \cdot v + B}{W} \right\rfloor$
k-dimensional
p-stable LSH bucket space
ξ_{sum}
Peer identifier space

Figure 10: Locality sensitive hashing (Haghani et al., 2008). Two level hashing from d-dimensional data space to peer identifier space.

Another set of techniques is based on a hybrid of spatial subdivision up to a threshold granularity and small scale brute force evaluation or heuristics for refinement (Dong et al., 2011; Chen et al., 2009; Wang et al., 2012). Some techniques take advantage of the intrinsic dimensionality of the data set to project the data set into a low-dimensional space that preserves proximity. These techniques use the results of the Johnson-Lindenstrauss theorem that says for any n point subset of Euclidean space can be embedded in k = $O(\log(n \times \varepsilon^2))$ dimensions without distorting the distances between any pair of points by more than a factor of 1 + ε for any $0 < \varepsilon < 1$. This reduces the complexity of the search. Examples of such work include techniques using random projections as in (Paredes et al., 2006).

2.3.2. High-dimensional data and curse of dimensionality

The main computational part of Diffusion Map is the construction of a *k*-NN distance graph for a high-dimensional data cloud. The input data to the algorithm contains upwards of 10^7 images, with each data point having up to 160k pixels.

For practical purposes, when one insists on having linear or near linear space requirement, the best performance time per query for a random input point cloud is

www.manaraa.com

bounded to $min(2^{O(d)}, d \times n)$, where d is data dimension and n is number of data points, which is essentially equal to *exhaustive search* (Indyk, 2004). In other words, the complexity of algorithms is linearly related to dimension and data number, even for moderate dimensions. Exponential dependence of time or space on dimension in *k*-NN search is termed *the curse of dimensionality* and has been observed in practical experiments. Many well known structures exhibit linear time search with linear or near linear storage even with moderate dimensions (10-20).

The curse of dimensionality leads to a belief supported by many researchers that the most efficient method for finding *k*-NNGs for high-dimensional data clouds is in fact the brute force method (Jing et al., 2012).

2.3.3. Brute force *k*-NN solutions

The brute force algorithm breaks into two parts: distance calculation and comparison. In the distance calculation part, all distances between all points for graph construction are computed. That results in an M × N distance matrix, where M is the number of query points and N is the number of data-base points. Next, each row of the matrix is sorted to get the nearest k neighbors to each of the query points. Fortunately, due to their simplicity, brute force methods are highly parallelizable and can be processed by computational clusters, clouds and high throughput parallel processors such as Graphical Processing Units (GPU).

المنارة للاستشارات

2.3.4. Implementation on Graphical Processing Units

Recently, there have been several methods that accelerate brute force *k*-NN and *k*-NNGs on graphics processing units. The complexity of the brute force algorithm brings the need for the development of algorithms and implementation on massive parallel processors such as GPUs. There are some GPU implementations of the brute force algorithm both for *k*-NN search and *k*-NNG construction.

(Garcia et al., 2010) proposed an algorithm for a *k*-NN search by devising two kernels for distance calculation based on cuBLAS (Volkov and Demmel, 2008), an optimized matrix multiplication library on the GPU, and a comparison kernel based on a parallel insertion sort, and achieved a 100-fold increase in speed compared with the approximate nearest neighbor (C++ ANN library).

This method works on multiple queries simultaneously, with each thread handling a single query. If *k* is small, then the data structure for an insertion sort can be stored in a fast on-chip shared memory and this method can be quite efficient. For a large *k* the insertion sort data structure spills into the main memory and causes a dramatic loss of efficiency because of un-coalesced memory transactions. In fact, for a large *k*, the selection is much slower than a simple sort operation.

(Kato el al., 2010) proposed a multi-GPU brute force *k*-NNG algorithm. Data are partitioned and distributed among 3 GPUs on a single computing cluster node. They use symmetry of the distance matrix to compute only half the entries. For each data partition called a grid. In the second phase, a heap-based selection is employed. Each row of the distance matrix is processed by a thread block. There is a per thread block heap that

المنارة للاستشارات

www.manaraa.com

stores the k smallest/largest elements. Each thread maintains a local buffer that stores the thread elements that are smaller/larger than the smallest/largest element in the heap.

Work presented in (Kato el al., 2010) uses a slightly different approach that (Garcia et al., 2010) to finding *k*-NN given the distance matrix. Here a single thread block handles each row of the distance matrix. Each thread stores a heap of *k* elements that records the local *k*-NN. Each thread strides through the given row of the distance matrix in a coalesced manner to find the local *k*-NN. At the end of this process, all threads in the block have their own heaps. In the next step, a single warp is used to build 32 *k*-NN heaps, one for each thread. In the final step, the first thread of the first warp reduces the 32 *k*-NN heaps to get the final *k*-NN. The second and third steps of this algorithm lose a large amount of parallelism and therefore underutilize GPU resources. Finally, this method works well only if k is small such that the thread heaps can be stored in an on-chip shared memory. Otherwise, just like the Garcia *k*-NN, the heap is stored in global memory that necessitates un-coalesced global memory reads.

In (Kuang and Zhauo, 2009) a radix sort-based approach is used to select the k nearest neighbors. The authors claim that for large data sets, especially for a large number of queries, the selection process dominates. A simple complexity analysis suggests that this is quite impossible ($O(d \times m \times n)$ for distance calculation vs. $O(m \times n \times log(n))$ for sorting). A closer examination shows that the approach process each row of the distance matrix in a separate sort. For an n that fits into GPU memory, this process underuses GPU resources.

www.manaraa.com

In another implementation of a brute force *k*-NN search algorithm, (Sismanis et al., 2012) proposed the truncated bitonic sort (TBiS), a selection method based on the biotonic sort. One of the main characterizations of the proposed method is having a low synchronization cost achieved by using synchronous memory operations. The TBiSort uses recursion to break down input arrays all the way to the base level and then goes upwards, merging the values. In the merging step, minimum and maximum values of each element pairs of two lists are detected and assigned to two minimum and maximum monotonic sub-lists. Having truncation, only the k elements of the minimum sub-list are gathered in each step and the maximum sub-list is set free in each step upwards. Truncation starts and continues from a minimum sub-list with k elements. While this method performs significantly faster than a plain sort and selects for small values of k and n, it rapidly loses efficiency as k and n grow. While it is 16x faster than a plain radix sort and selects for $k = 2$ and $n = 2^{17}$, for $k = 2^8$ and $n = 2^{20}$ it is no better than a plain radix sort and select.

2.4. Distributed parallel processing on GPU cluster

The *k*-NNG implementation developed in this thesis uses multi-level parallel execution on computing clusters with GPU accelerators. In this section the various tools, techniques and programming models used in this implementation are described.

المنارة للاستشارات

www.manaraa.com

2.4.1. Massage passing interface (MPI)

Message passing interface (MPI) is the most widely used application protocol interface (API) for distributed memory parallel execution on computing clusters. It is a message-passing application programmer interface, together with protocol and semantic specifications for how its features must behave in any implementation (Pacheco 1997). It provides language binding for C, C++ and FORTRAN, and currently is the de facto standard interface for high performance and high throughput computing applications on distributed memory architecture. In the MPI programming model one node in the computing cluster is designated as the head node and controls the overall execution. Other nodes are designated as worker nodes and are responsible for the distributed execution. MPI provides point-to-point message passing for user-specified groups of exclusive processes. In programming with MPI, worker nodes access each others’ data through the high-speed network connecting the cluster. In addition, there is an option for each node to broadcast its data to all other worker nodes in the cluster.

2.4.2. Open multiprocessing (OpenMP)

In contrast to MPI, OpenMP is a shared memory architecture API. OpenMP enables the harnessing of multiple cores on modern CPUs through multi-threading. OpenMP follows a fork-join programming model, where a parent task/process can spawn multiple tasks that can execute in parallel. Parallel tasks can share data and synchronize. Tasks can execute entirely different programs on different data types. In the OpenMP task execution model each task can have independent access to memory and cache. OpenMP supports

المنارة للاستشارات
www.manaraa.com

bindings for popular programming languages such as C, C++ and FORTRAN. In a cluster setup, MPI can be used for assigning the tasks to worker nodes and multithreaded task executions inside the nodes can be achieved with the help of OpenMP.

2.4.3. Graphics Processing Units (GPUs)

Graphics Processing Units were initially developed to handle computations related to graphics rendering. The need for specialized rendering routines led GPU vendors to provide user-defined functionality through shader programming (Luebke et al., 2006). Subsequently, researchers used shaders to essentially trick GPUs into performing scientific computations. This further led GPU vendors to develop extensions of the C language in order to directly access the parallel processing power for general purpose scientific computing. Examples of these APIs include CUDA (Sanders and Kandrot, 2010), OpenCL (Khronos, 2008), and OpenGL (Woo et al., 1999).

The processing elements in a GPU are organized around several multi-processors (MPs). Each multi-processor, as the name suggests, has several serial processing units (SPUs). All SPUs in an MP have access on-chip to a user-controlled cache called shared memory. In addition, there is a register file that is distributed among all MPs. Shared memory is organized into memory banks. The off-chip RAM on a GPU is called global memory. While previous generation GPUs did not cache global memory, the latest generation GPUs have L1 and L2 caches. L1 cache is local to an MP while the L2 cache is shared among all MPs (Figure 11). A small portion of the global memory is designated as read-only constant memory. This memory that is automatically cached can be used to store constant values that are frequently accessed.

المنارة للاستشارات

www.manaraa.com

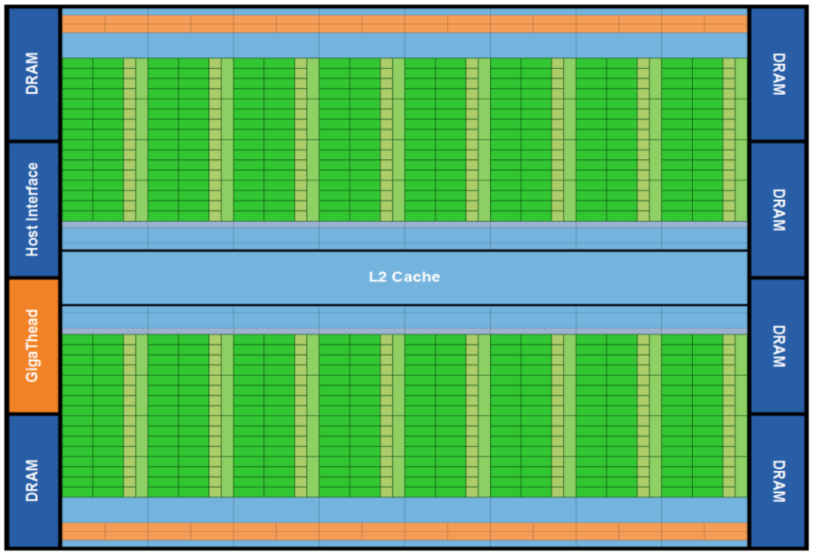


Figure 11: GPU Fermi architecture (Sanders and Kandrot, 2010). 16 multiprocessors are positioned around a common L2cache. Each orange portion is scheduler and dispatch, green portions are execution units andlight blue portions are memory for register file and L1 cache

GPUs generally follow the data-parallel programming model, where the same instruction is applied to elements of a data array. For GPUs developed by NVIDIA, there is a native API called Compute Unified Device Architecture (CUDA), which dramatically decreases the programming overhead involved in accessing the parallel computing power of GPUs. The basic execution unit is a thread. Every thread executes the same program called a kernel. Threads are organized into logical partitions called thread blocks (TBs). During execution, all threads in a TB are assigned to a single MP. Threads in a TB can communicate via shared memory and can be synchronized. In a typical execution, the number of TBs far exceeds the number of MPs (Figure 12). At the hardware level, threads are organized into warps. All threads in a warp execute in lock-step fashion. Warps are equivalent to threads in the symmetric multi-process context. It is therefore advisable to avoid thread divergence with a warp. If two or more threads in a warp access

www.manaraa.com

the same shared memory bank, this operation will cause bank conflicts and serialization. However, if all threads access a single shared memory location, then an efficient broadcast mechanism is used. In accessing global memory, all threads in a warp must access memory within a contiguous 128b segment. Non-compliant memory accesses are called un-coalesced accesses and are serialized.

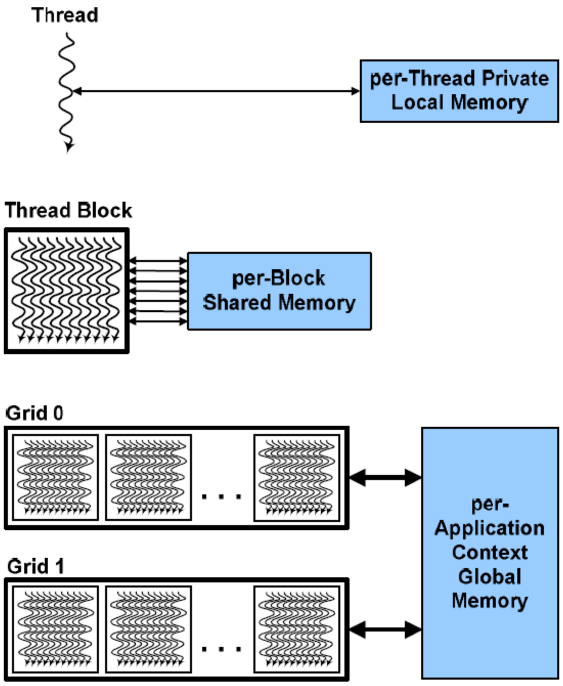

Thread
per-Thread Private Local Memory
Thread Block
per-Block Shared Memory
Grid 0
Grid 1
per-Application Context Global Memory

Figure 12: CUDA Model (NVIDIA, 2008). Threads, blocks, and grids, with corresponding memory spaces for private per-threads, shared per-block, and global per-applications.

2.5. Algorithm development

In this section, an expandable method for the construction of nearest neighbor graphs for very large high-dimensional data clouds is presented using the brute force method. The

www.manaraa.com

method presented in here relies on three levels of parallelism, is designed to execute on computing clusters with GPU accelerators in the nodes, and addresses the computational complexity of the brute force method. The three levels of parallelism are: distributed memory parallelism between the nodes of the cluster; shared memory parallelism between the cores of the CPUs in the node; and finally, data-parallelism within the GPU accelerators in each node. The main contributions of this thesis are the following: (a) A novel scheme for data partitioning and management for load balancing between the nodes of the cluster and efficient communication. (b) Two algorithms for selecting *k*-NN on GPUs. The first is based on indexed sorting and relies on the optimized radix sort algorithm available in the Thrust library (Bell and Hoberock, 2011). The second is an efficient implementation of the quick-select algorithm by using latest GPU functionalities. To our knowledge, the last algorithm is the fastest multi-query *k*-NN select algorithm to date.

2.5.1. Distributed *k*-nearest neighbors graph construction on GPU clusters

Brute force methods have two primary tasks, namely, generation of the distance matrix between input vectors, and selection of *k*-NNs. Both these tasks are computationally expensive. Due to the massive input data size and the even more massive intermediate results (distance matrix), we use a distributed approach with interleaving of distance calculation and *k*-NN selection along with merging of results. In this section, we begin by describing the distance calculation. Next, we describe our data partitioning approach to handle the massive memory footprint as well as to balance the computation. Finally, we describe the two new algorithms for *k*-NN selection.

المنارة للاستشارات

www.manaraa.com

In this thesis we focus on calculating the Euclidian distance. A similar approach can be taken for other distance metrics such as Cosine or Pearson distance. Given two D-dimensional vectors v_i; v_j , the square of the Euclidian distance is given by:

$$d(v_i, v_j) = \sqrt{\left\|v_i - v_j\right\|^2}$$

$$d^2(v_i, v_j) = \left\|v_i - v_j\right\|^2$$

$$= \left(v_i - v_j\right)^T \left(v_i - v_j\right)$$

$$= {v_i}^T v_i + {v_j}^T v_j - 2{v_i}^T v_j$$

$$= \|v_i\|^2 + \left\|v_j\right\|^2 - 2{v_i}^T v_j$$

Now consider a set V = [v_1 v_2 ... v_n]. Further consider a squared distance matrix that contains the mutual distance between all vectors in V:

$$S = \begin{pmatrix} d^2(v_1, v_1) & d^2(v_1, v_2) & \dots & d^2(v_1, v_n) \\ d^2(v_2, v_1) & d^2(v_2, v_2) & \dots & d^2(v_2, v_n) \\ \dots & \dots & & \dots \\ d^2(v_n, v_1) & d^2(v_n, v_2) & \dots & d^2(v_n, v_n) \end{pmatrix}$$

Defining matrices $A^{N \times D}$, $B^{N \times N}$ be given by

$$A = \begin{pmatrix} v_1^T \\ v_2^T \\ \dots \\ v_N^T \end{pmatrix}$$

$$B = \begin{pmatrix} \|v_1\|^2 & \|v_1\|^2 & \dots & \|v_1\|^2 \\ \|v_2\|^2 & \|v_2\|^2 & \dots & \|v_2\|^2 \\ \dots & \dots & \dots & \\ \|v_N\|^2 & \|v_N\|^2 & \dots & \|v_N\|^2 \end{pmatrix}$$

We can now write the following equation

www.manaraa.com

$$S = B + B^T - (2AA^T)$$

Therefore, the calculation of the squared distance matrix can be formulated in terms of vector reductions, vector additions, and dense matrix multiplication. All of these are BLAS routines and have very efficient libraries on GPUs. Note that the S is symmetric and therefore, it is enough to compute only elements $S(i,j) \mid j \leq i$. Finding the set of the k nearest neighbors for vector vi involves sorting the i_{th} row of S and picking the column indices corresponding to the k smallest distances. To handle the large data size, our approach is to compute the k nearest neighbors in parts. As illustrated in Figure 13, computation of the squared distance matrix S is split into P×P partitions. Consequently, each portion $S(I,J)$ is computed as:

$$S_{(I,J)} = B_I + B_J^T - 2\left(A_I A_J^T\right)$$

where A_I ;B_I I = 1;2...P are partitions of A;B respectively.

www.manaraa.com

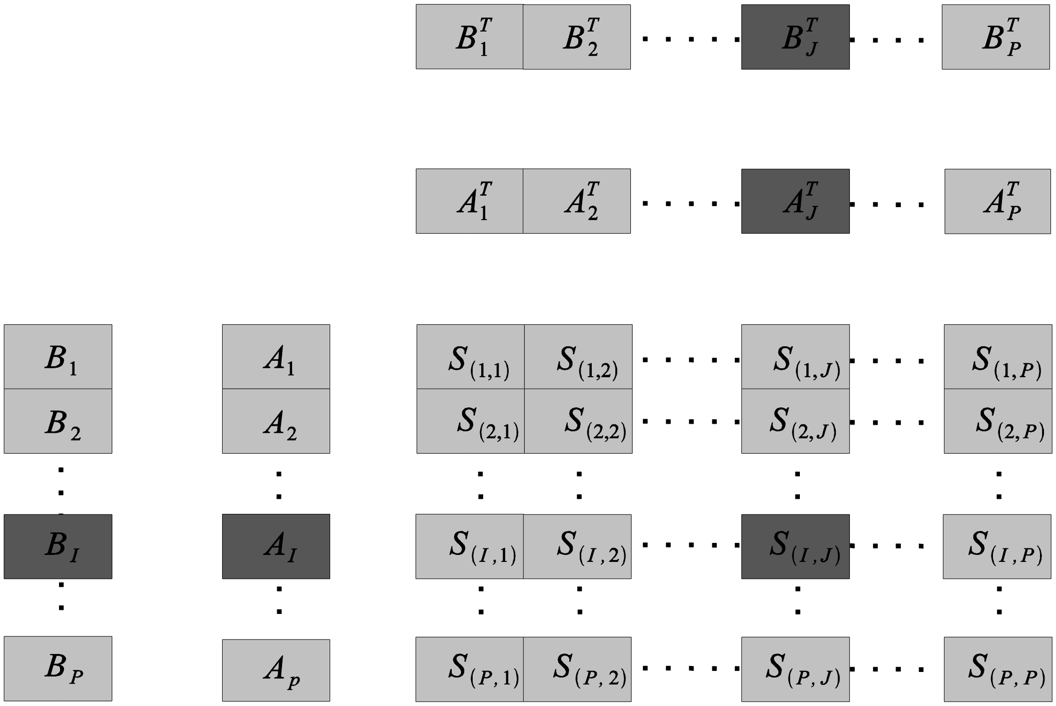


Figure 13 Data partitioning for distributed parallel execution.

Computing clusters typically have several nodes connected by high-speed interconnects. One of the nodes is designated as the head node, which typically co-ordinates the tasks between different worker nodes. Each node has its own hard disk. In addition, there is a large shared disk accessible by all nodes through parallel (I/O) that typically holds input data and results. The worker nodes, with smaller local disk space, copy input data from the shared disk as required.

For load balancing, we distribute computing of the partitions of S in a block cyclic manner. This means that node q computes all the partitions $S_{I,J}$ | J : J%Q = q. Now

المنارة للاستشارات

consider the case where the I_{th} block row of S is being processed. Any block $S_{I,J}$ requires inputs A_I ;A_J ;B_I ;B_J . Of these, A_I , B_I are used by all nodes that are processing the I_{th} block row.

Each node q also requires B_J , A_J | J : J%Q = q. The disk space on the nodes restricts the number of partitions of A that can be saved locally. Therefore, in our setup, the vector data A is uploaded on the shared disk and divided into A_1,A_2...A_P partitions. The partition AI is read in parallel from the shared disk on the head node while partitions A_J| J : J%Q = q are stored locally on node q. AI is then read in parallel by all nodes from the shared disk and then the portions are shared by using an asynchronous ‘all gather’ operation to build an image of AI in each node's RAM. Figure 14 illustrates this process. We use message passing interface (MPI) to distribute the computational tasks as well as to communicate data between various nodes in the cluster.

We do not actually build the matrix B. Instead, the vector $\hat{B} = \{\|v_1\|^2, \|v_2\|^2, \dots, \|v_N\|^2\}^T$ is computed in advance and stored in the RAM of each node. Even for $N = 10^7$ the size of B (> 10MB) is quite small compared with the RAM in each node (48 GB). Each node q computes the vector norms for all vectors in the partitions A_J | J%Q = q resident on its disk space locally. It then broadcasts the results to all other nodes.

www.manaraa.com

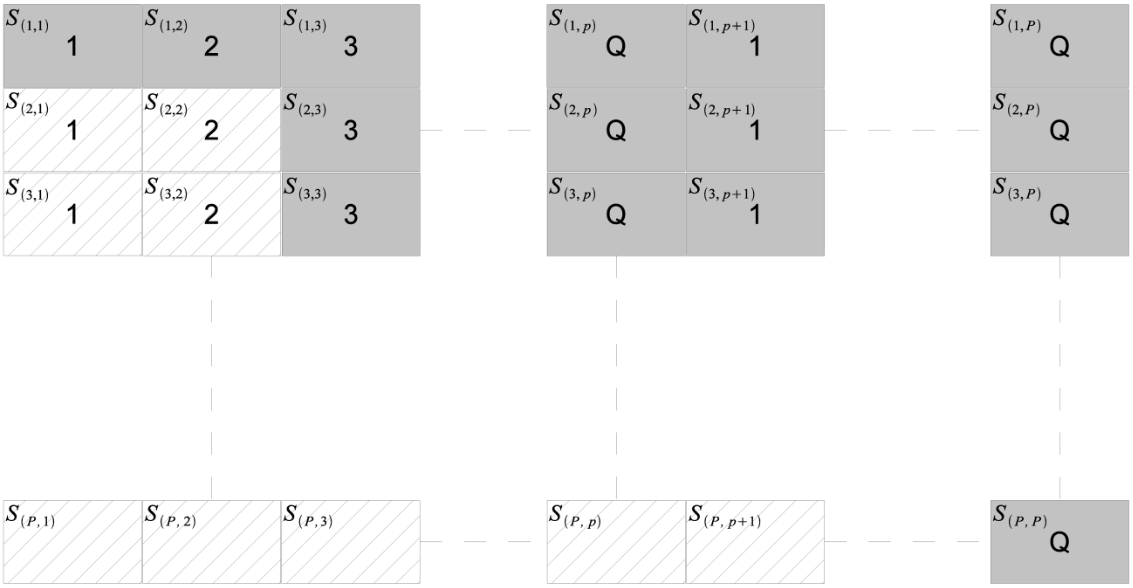
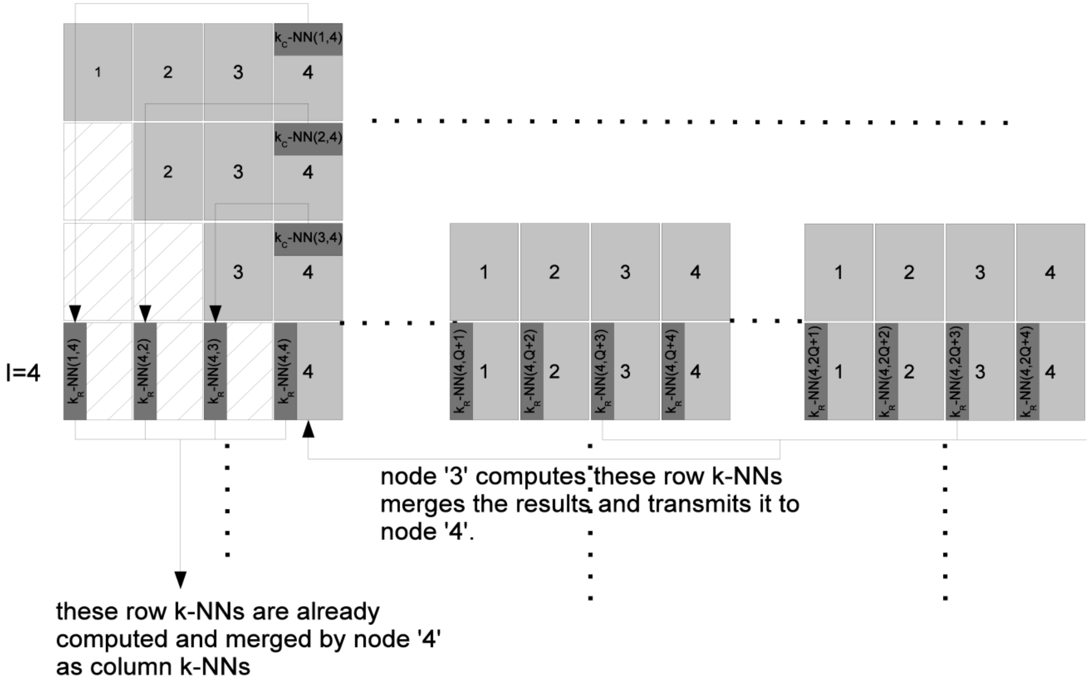


Figure 14: ***Load balancing between cluster nodes.***

Once the partition $S_{(I,J)}$ is computed, the local *k*-NNs with respect to both the rows (k_R-*NNs*) and columns (k_C-*NNs*) are computed. Since the matrix *S* is symmetric, the local k_C-*NNs* w.r.t. partition $S_{(I,J)}$ are identical to the local k_R-*NNs* w.r.t. partition $S_{(J,I)}$. Therefore, each node q maintains one heap per column J | J%Q = q of *S* that it processes. Each of these heaps contains the merged local k_C-*NNs* w.r.t. partitions $S_{(I,J)}$ | I = 1, 2, ... J, J%Q = q. For example, as shown in Figure 15, node 4 maintains one heap for each of the columns 4,Q+4, ... (P-Q+4). The heap for column 4 will contain the merged local k_C-*NNs* for $S_{(1,4)}$, $S_{(2,4)}$, $S_{(3,4)}$, $S_{(4,4)}$. The heap for column Q + 4 will contain the merged local k_C-*NNs* for partitions $S_{(1,Q+4)}$, $S_{(2,Q+4)}$, ..., $S_{(Q+4,Q+4)}$.

The node q is used to compute the global *k*-NNs for all vectors in A_I | I%Q = q. The global *k*-NNs for all the vectors in AI are generated by merging the local k_R-NNs w.r.t. partitions $S_{(I,J)}$ $J = 1, 2, ..., P$. However, the merged results of the local k_R-NNs w.r.t all partitions $S_{(I,J)}$ | $J = 1, 2, ..., I$ are already available in node q from the local k_R-NNs computed previously. The local k_R-NNs w.r.t. all partitions $S_{(I,J)}$ J = I + 1, I + 2, ... , P are cooperatively computed by different nodes. Each node maintains a heap to merge the results of finding the local k_R-NNs of the partitions that it processes.

At the end, the merged results are communicated to the node processing the global *k*-NNs for the block row I for merging at the global level. For example, for I = 4, node 3 will compute local k_R-NNs w.r.t. all partitions $S_{(4,Q+3)}$, $S_{(2,2Q+3)}$, ... , $S_{(4,Q-P+3)}$. The results will be merged and stored in a heap. At the end of the computation, the results in the heap will be communicated to node 4 for computing the global *k*-NNs for all vectors in A_4.



node '3' computes these row k-NNs merges the results and transmits it to node '4'.

these row k-NNs are already computed and merged by node '4' as column k-NNs

Figure 15: Computing k-NN in GPUs.

We assume that each node has M GPUs. In our current setup, M = 2. As mentioned previously, each node is responsible for computing a partition S(I,J) of the squared

www.manaraa.com

distance matrix. A_I s are read from the head node through parallel I/O. A_J s are read from the local disk. Within the node, AI is divided into M equal partitions. A_J is divided into R partitions. R is governed by the available GPU RAM. While reading A_I is quite fast (it is parallelized), reading A_J from the local disk is slow. We therefore hide this latency by reading the disk in parallel with computation.

Each GPU is given a partition of A_I denoted by A_I (m) | m = 1, 2, ... , M. A partition of the file A_J denoted by A_J (r) | r = 1, 2,...,R is read by all GPUs. When the computation of $S_{(1,J)}(1,r)$, $S_{(2,J)}$ $(2,r)$, ..., $S_{(1,J)}(m,r)$ is being done by the M GPUs, the node simultaneously reads the file partition A_J (r+1) into the RAM. The node also has the norm vector $\hat{B}$ in its RAM. $\hat{B}$ is partitioned in two ways: one has M partitions with each of these partitions going to the M different GPUs and the other has R partitions, with each partition being read sequentially by all GPUs. When the computation of $S_{(I,J)}(m, r)$ is complete, the column *k*-NNs as well as the row *k*-NNs are computed using sorting. The column *k*-NNs are written back to CPU memory while the row *k*-NNs are kept on global memory to be merged. Note that $S_{(I,J)}(m, r)$ r = 1, 2,...,R are being processed by the same GPU m, therefore, it makes sense to merge row *k*-NNs in the GPU memory without write back to CPU RAM. However, $S_{(I,J)}(m, r)$ m = 1, 2, ..., M are being processed by different GPUs, therefore, the column *k* -NNs are generated by different GPUs. The accumulated column *k*-NNs are then merged by one GPU per column. The final result of this operation is the local row and column *k*-NNs for the partition $S_{(I,J)}$. Figure 16 illustrates this process.

المنارة للاستشارات

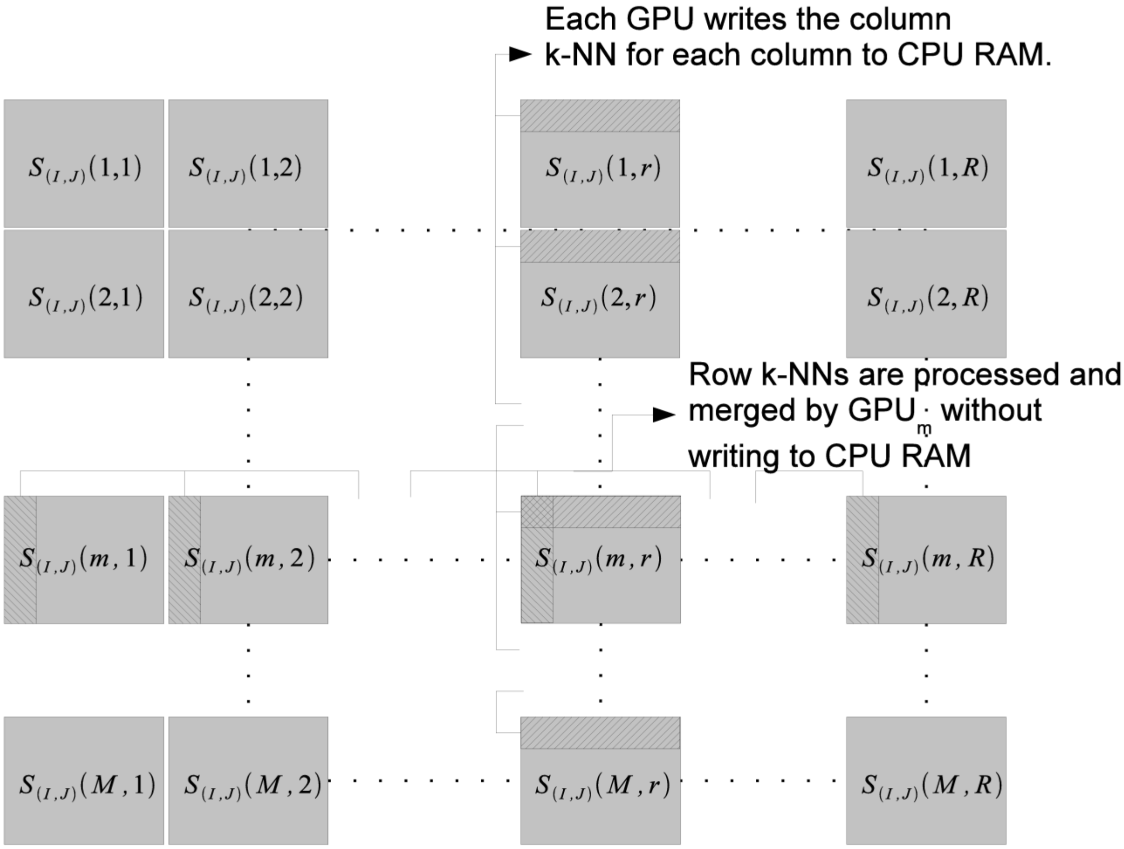


Figure 16: Task distribution within the nodes.

We use OpenMP multi-threading. Each node runs M+1 CPU threads. M threads control the M GPUs while one thread is in charge of I/O from the local disk as well as the shared disk.

The tasks that are accomplished within each GPU include the following:

- Finding vector $\hat{B}$ of input data norms
- Dense matrix multiplication to generate the result $2A_I(m)A_J^T(r)$
- Summation to find the result $S_{(I,J)}(m,r) = B_I(m) + B_J^T(r) - 2A_I(m)A_J^T(m)$

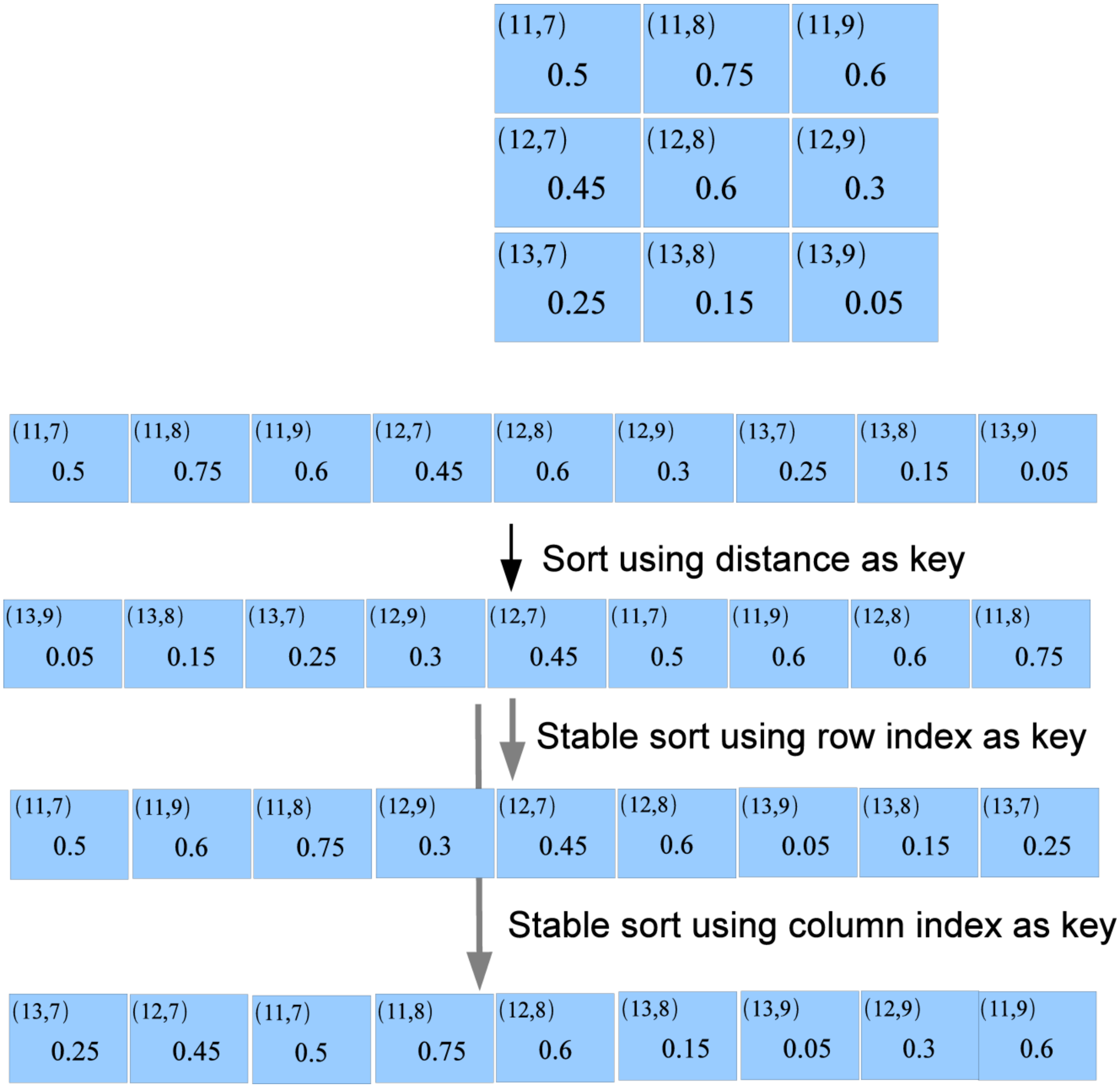
www.manaraa.com

- Finding local *k*-NNs based on $S_{(I,J)}(m, r)$

Finding vector $\hat{B}$ of input data norms is done once in the beginning at the same time that the input files A_J are communicated to each node. For example, if node q will receive all files $A_J \mid q = J\%Q$. When the file is received, it is partitioned into M partitions, one partition per GPU. Although there is a library function to calculate vector norms in CUBLAS (NVIDIA, 2008), using it will be inefficient since the vectors are relatively large in number ($N=10^6$ -10^7) with much smaller dimension $D \approx 15000$.

2.5.2. Batch index sorting

We could obviously sort each column and each row separately. However, this is not efficient because the resources on the GPU are not fully used. Also, for sorting according to columns, we will need to execute an expensive operation to rearrange the data in the column major format. Instead, we use a process we called Batch Index Sorting. Note that in GPU RAM, $S_{(I,J)}(m, r)$ is laid out as a linear array in a row-major format. Each element of $S_{(I,J)}(m, r)$ is also then associated with its row index and column index. We use radix sort with the elements of $S_{(I,J)}(m, r)$ as key. In the next two steps, we execute an order preserving sort of the result of the previous step with the column index as the key. Separately, we also execute an order preserving sort with the result of the first sort with the row index as the key. These two sorting operations generate the nearest neighbors for each column and row. We then execute a separate kernel to extract the *k*-NNs for each row and column w.r.t. $S_{(I,J)}(m, r)$. Figure 17 illustrates this process.



(11,7) 0.5	(11,8) 0.75	(11,9) 0.6
(12,7) 0.45	(12,8) 0.6	(12,9) 0.3
(13,7) 0.25	(13,8) 0.15	(13,9) 0.05

(11,7) 0.5	(11,8) 0.75	(11,9) 0.6	(12,7) 0.45	(12,8) 0.6	(12,9) 0.3	(13,7) 0.25	(13,8) 0.15	(13,9) 0.05

Sort using distance as key

(13,9) 0.05	(13,8) 0.15	(13,7) 0.25	(12,9) 0.3	(12,7) 0.45	(11,7) 0.5	(11,9) 0.6	(12,8) 0.6	(11,8) 0.75

Stable sort using row index as key

(11,7) 0.5	(11,9) 0.6	(11,8) 0.75	(12,9) 0.3	(12,7) 0.45	(12,8) 0.6	(13,9) 0.05	(13,8) 0.15	(13,7) 0.25

Stable sort using column index as key

(13,7) 0.25	(12,7) 0.45	(11,7) 0.5	(11,8) 0.75	(12,8) 0.6	(13,8) 0.15	(13,9) 0.05	(12,9) 0.3	(11,9) 0.6

Figure 17: Batch index sorting.

2.5.3. Quick select

The quick select algorithm is a variant of the quick sort algorithm. Given an array, just as in the quick sort algorithm, a random element is selected as the pivot. Next, the array is re-arranged with the elements less than the pivot being moved to the left of the pivot and

المنارة للاستشارات

www.manaraa.com

the elements greater than the pivot being moved to the right side. This process is recursive on the right partition and the left partition until the partition size reaches two elements and these elements can be swapped. In the quick select algorithm, as soon as the partition is finished, the sizes of the left and right partitions, L and R respectively, are found. If L > k, then the right partition is discarded and the left partition is further recursively processed. If L < k, then the left partition is kept. Next we set k = k - L. Then the right partition is processed recursively. This algorithm has complexity $O(n) + k \log(k)$ for a sorted *k*-NN list and $O(n)$ for the unsorted *k*-NN list.

Our approach operates on multiple arrays simultaneously. Each array is handled by a single thread warp. Threads in a warp are executed simultaneously on a single multi-processor and therefore are synchronized by default. Partitioning of an array is done incrementally in 32-element wide segments. We use shared memory to ensure coalesced memory writes of the results of partitioning into the auxiliary array in global memory. Our approach uses the warp voting function *ballot(p)* to partition the input without reading the input array twice and without executing the parallel-prefix sum. The ballot function *ballot(p)* fills a 32-bit unsigned integer, one bit per thread in the warp, based on the evaluation of the predicate p.

Each thread in the warp first reads in an element into its register from global memory. Elements are then written to a 32-element wide shared memory array with elements greater than equal to the pivot being written from the right end and elements less than the pivot being written from the left end. To do this, each thread needs to know where in shared memory to off load

www.manaraa.com

its element. We execute the warp voting function based on a predicate that checks whether the element in the register is greater than or equal to the pivot or smaller than the pivot. Threads that have an element greater than or equal to the pivot set the corresponding bit to ’1’ and to ’0’ if the element is less than the pivot. This process is illustrated in *Figure 18*.

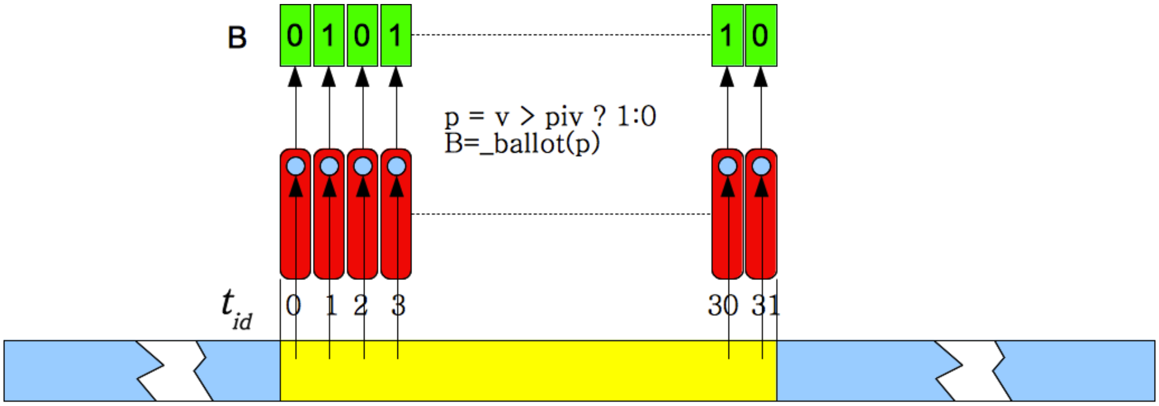
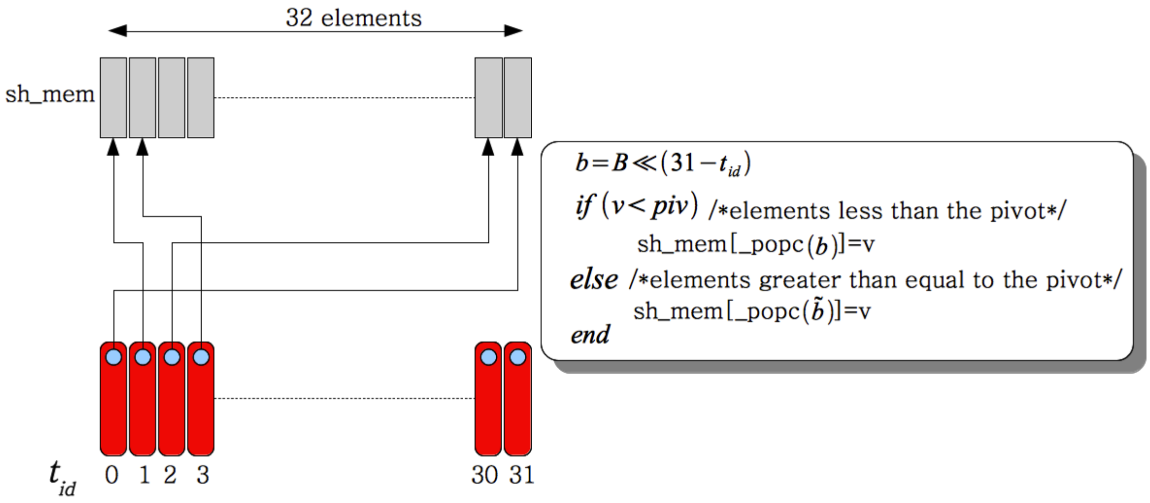


Figure 18: Read in process. Read in of the array is done incrementally in sets of 32 elements. As illustrated, the memory access is coalesced. The value is stored in a register. Simultaneously, the invocation of the warp voting function fills the bit array B based on the evaluation of the predicate that indicates if value in the register is greater than, equal to or less than the pivot.

If the element in the register is less than the pivot, then the thread needs to find how many of threads before it have elements less than the pivot and vice-versa. We use a combination of bit shift operations and the *popc(x)* function on the integer result of the warp voting to accomplish this. The *popc(x)* functions count the number of bits set to ’1’ in the input integer ’x’. For example, if the warp vote integer in binary is B = 0101..., then it is clear that threads 0,2 have elements less than the pivot and threads 1,3 have elements greater than or equal to the pivot. Each thread i computes the result $b = B >> (31 - i)$. When the *popc(b)* function is applied to the result of this step, it will indicate the

www.manaraa.com

position in shared memory at which thread i will off-load its element that happens to be less than the pivot. Similarly, *[31 - popc(~b)]* will indicate the position from the right side at which thread i will off-load its element that happens to be greater than or equal to the pivot. In the example, thread 3 will bit shift B to the right by 29 bits and the result will be b = 0101. Then *popc(b)=2*. Therefore, thread 3 will off-load its element at the second location from the left in shared memory (Figure 19).



```
$b = B \ll (31 - t_{id})$
if ($v < piv$) /*elements less than the pivot*/
    sh_mem[_popc($b$)]=v
else /*elements greater than equal to the pivot*/
    sh_mem[_popc($\tilde{b}$)]=v
end
```

Figure 19: Pivot process. The pivot process is accomplished in shared memory. Each thread determines where in the shared memory the value has to be written. Values less than the pivot are accumulated on the left hand side and values greater than or equal to the pivot are accumulated on the right hand side. Since all threads write to different locations, there are no bank conflicts.

Note that the total number of elements in shared memory that are less than the pivot is given by *popc(B)*. Two global counters, *g<* and g≥, keep track of the total number of elements less than the pivot and the total number of elements greater than or equal to the pivot, respectively, are also maintained. These two counters indicate the location in the auxiliary array at which the warp writes the incremental results of pivoting from shared memory. Next, the threads write the contents of the shared memory into the global

www.manaraa.com

auxiliary array with threads whose id is less than *popc(B)* writing from the left side and other threads writing from the right side. This write process requires two coalesced writes, one for elements smaller than the pivot and one for elements greater than or equal to the pivot, into the auxiliary array (*Figure 20*).

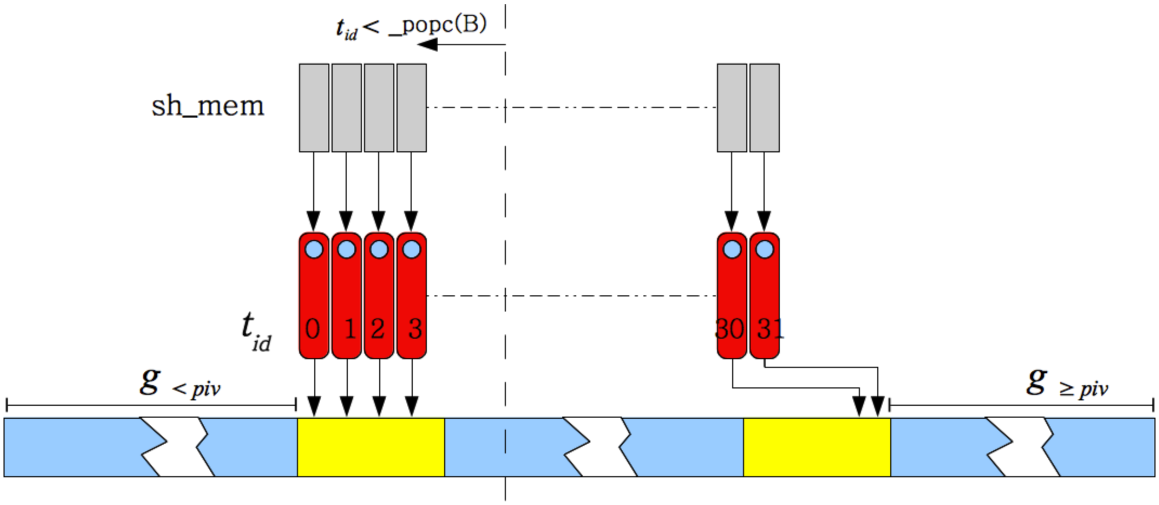


Figure 20: Write-out process. The thread id indicates (based on the computation *popc(B)* whether a given thread is writing out an element less than the pivot or greater than or equal to the pivot. The values g< piv and g≥ piv that are maintained in shared memory and updated incrementally indicate the location in the global array the location of the last element that is less than the pivot and greater than or equal to the pivot. This operation involves at most two coalesced memory writes.

Once a partition is complete, the output array (auxiliary array) has a left side of length L elements, each of whom is less than the pivot. The right side is of length R elements, each of whom is greater than equal to the pivot. Suppose we need k nearest neighbors, and k < L; then we need to process only the left hand side. Suppose k > L; then we keep the left hand side as is, and partition the right hand side to find k - L elements. Since the input and auxiliary arrays are swapped at the end of the partition process, in the second case (k

www.manaraa.com

> L), we would have to copy the left hand side from the auxiliary array to the input array. We can avoid copying the data by storing a stack of references that indicate the start and end indices and the arrays (auxiliary or input) where the partitions that form the k nearest neighbors are to be found. This significantly reduces the memory transactions needed for the operations.

2.6. Performance benchmarks

In this section we present the results of our benchmarks. Our implementation of task distribution and data partitioning is unique to the problem at hand; i.e., calculating one half of the distance matrix and generating *k*-NNG. While there are optimized algorithms for dense matrix multiplication on distributed computing systems, these methods are not suitable for our application. However, there are several comparable exact brute force *k*-NN implementations on GPUs (Garcia et al., 2010; Kato and Hosino, 2010; Arefin et al., 2012; Barrientos et al., 2011). We chose to benchmark against (Garcia et al., 2011) and (Arefin et al., 2012), since the code is readily available. All implementations were compiled using C++ with appropriate compiler optimization flags. The implementations were run on a NVIDIA Tesla C2050. We used synthetic data sets for performance analysis. We provide benchmark results for distance computation, *k*-NN selection, and total calculation time.

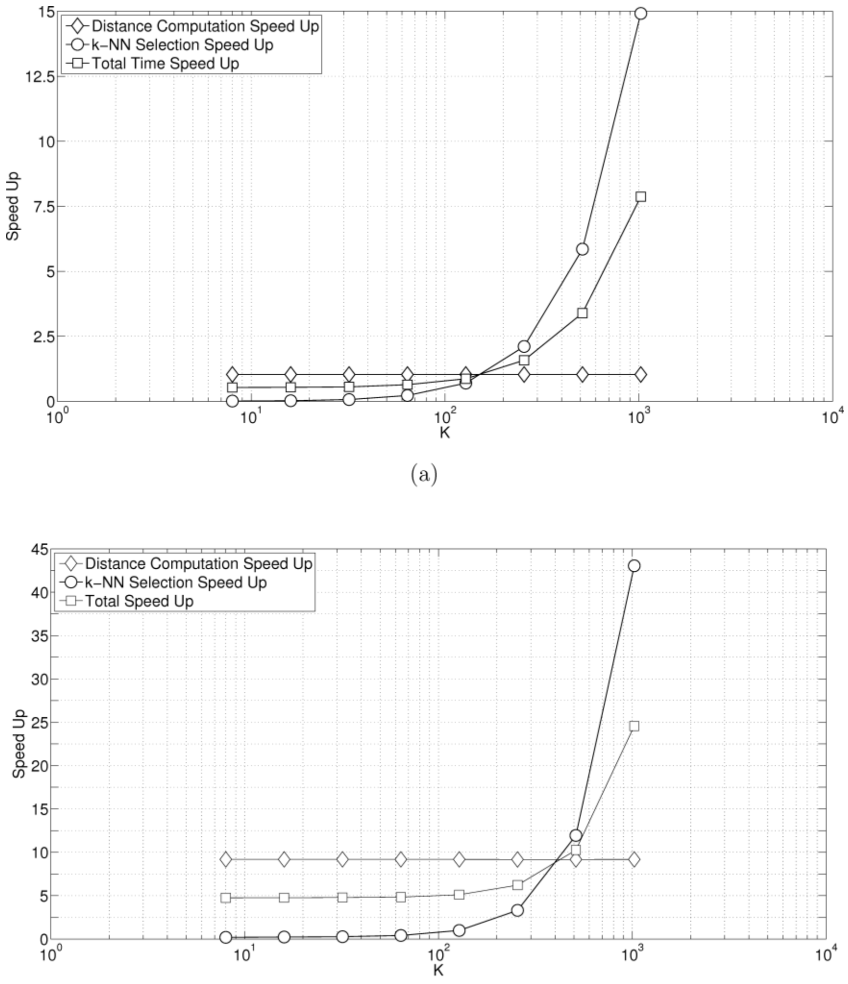
www.manaraa.com

2.6.1. Batch index sorting

This section presents performance analysis of *k*-NNG with the batch-index sorting algorithm. First the results of single GPU benchmarks are presented. Performance analysis was performed for varying different parameters, which showcase the behavior of our implementation for different scales. Benchmarks were obtained with various k (number of wanted nearest neighbors), n (number of data points) and D (dimension of each data vector). Our performance benchmarks show that our algorithm performance is superior to those of Garcia and Arefin, even in single GPU execution. In addition, we benchmarked multi GPU performance analysis vs. that of Arefin, reaching better performance due to proper task distribution with a symmetric *k*-NNG structure.

Figure 21 compares the performance of our first *k*-NNG algorithm with that of Garcia with varying k. In this test two constant parameters, data dimension and the number of samples, were set to d = 4096 and n = 16384, respectively. Figure 21*(*a) shows the distance calculation, selection and the total *k*-NNG speedup. The distance computation time is almost the same for both algorithms, because both formulate distance computation as a matrix-matrix multiplication and use the optimized CUBLAS library. Our version of the *k*-NN selection breaks even with the work of Garcia at about k = 128 and outperforms it by 15× for k = 1024. Overall, our implementation has a performance advantage of 7.87×.

www.manaraa.com



(a)

(b)

Figure 21: Performance analysis of batch index sorting (varying k). benchmark results for varying k in comparison with (Garcia et al., 2010) and (Arefin et al., 2012). In this test our input data has the dimension d = 4096 and the number of input objects/vectors n = 16384. (a) shows the performance vs. Garcia. (b) shows the performance vs. Arefin.

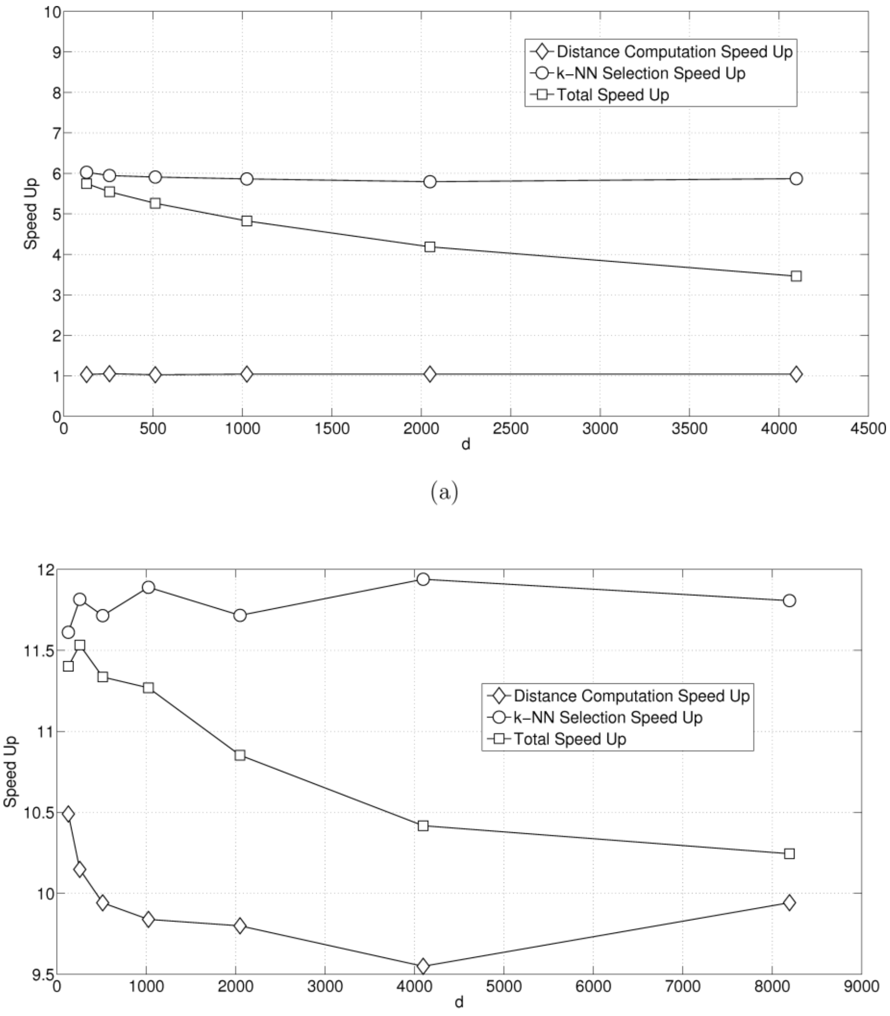
In Figure 21 (b) we re-formulated the Pearson distance computation to enable the use of optimized matrix-matrix multiplication. Consequently, our distance implementation has a roughly 9× performance advantage. The *k*-NN implementation is a per-thread linear insertion sort with each thread handling one row of the distance matrix. Our implementation breaks even at k = 128 and ends up with a 42× performance advantage when k = 1024. Overall, our implementation has a 24× performance gain at k = 1024.

The degradation of performance of the *k*-NN selection algorithms in Garcia and Arefin occurs at large values of k, because the temporary list of k elements maintained for each thread does not fit into the fast shared memory on GPUs and is therefore maintained in global memory. Because of the nature of the insertion sort and the heap sort, this causes thread divergence and un-coalesced memory transaction that results in a huge drop in performance.

In order to show the performance of *k*-NNG with batch index sorting for different data dimensionalities, we performed a second set of benchmarks in which n = 16384 and k = 512 and d was varied. Figure 22 (a) shows the comparison with Garcia. As the complexity of distance calculation is linearly related to the number of data dimensions and the complexity of the selection part of *k*-NNG is not related to the data dimensionality, the speedup with respect to both distance and *k*-NN selection remains constant at 1× and 6×, respectively. However, as the proportion of time for the distance computation increases linearly with d, the performance gain for the total time decreases from 5.7× to 3.5×.

Figure 22(b) shows the speedup with respect to Arefin. Once again, the speedup with respect to distance and with respect to *k*-NN selection remains constant at 9.5× and 10×, respectively. As d increases, the proportion of time for the distance computation increases linearly with d. For the large d, the graph shows the stabilization of the overall speed up at 10.25×.

www.manaraa.com



(a)

(b)

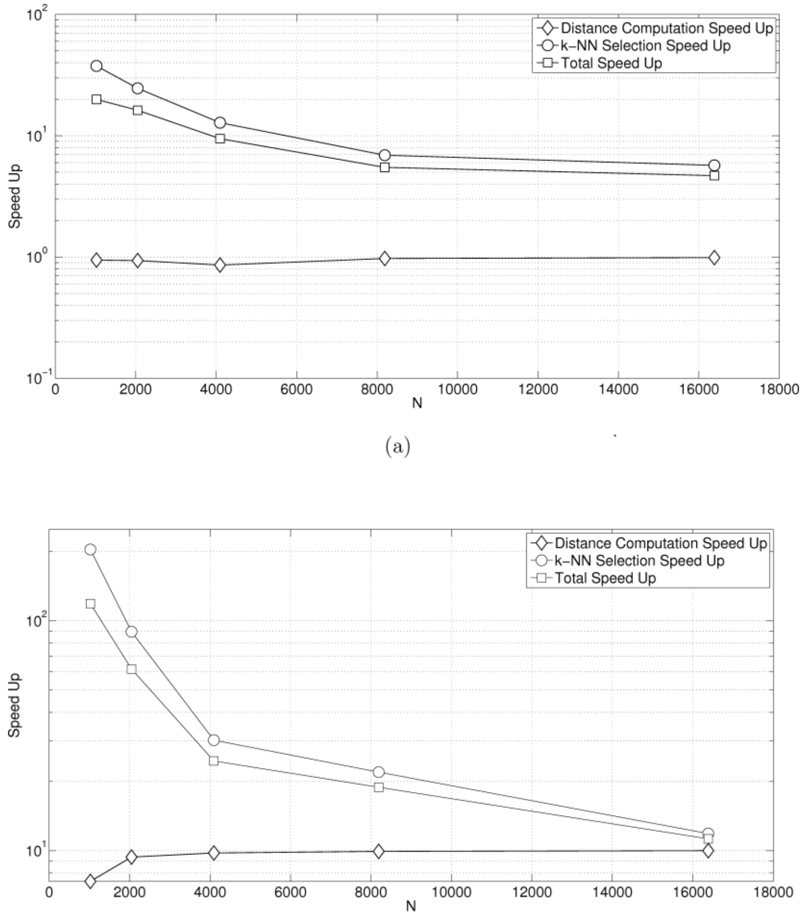
Figure 22: Performance analysis of batch index sorting (varying d). Benchmarks for varying d. In this test our input data has the number of closest neighbors k = 512 and the number of input objects/vectors n = 16384. (a) shows the performance vs. Garcia. (b) shows the performance vs. Arefin.

Next we benchmarked the performance for different values of n. Specifically, we kept d = 1024 and k = 512 and varied n. Figure 23 shows the comparisons. For a small n, the speedup with respect to selection is 200× against Garcia. As n increases, the performance gains taper off to 12×. Overall speedup starts off at 100× and falls to 11×. Figure 23 (b)

المنارة للاستشارات

www.manaraa.com

shows the comparison with Arefin. Once again, for a small n, the speedup with respect to selection is ≈ 37×. As n increases, the speedup tapers off to 5.6×. Overall speedup starts off at 20× and tapers off to 4.7×. Finally, for the tests with varying d and n, while our implementation was able to handle a model size up to n = 32, 767, the implementations by Garcia could only handle a model size up to n = 16, 384. Note that our *k*-NN method grows proportional to n^2 *log(n)* as opposed to n^2 *log(k)*. However, for data with the ranges of n that fit into GPU memory, our *k*-NN method is still much faster.



(a)

(b)

Figure 23: Performance analysis of batch index sorting (varying n). In this test our input data have the dimension k = 1024 and the number of input objects/vectors d = 4192. (a) shows the performance vs. Garcia. (b) shows the performance vs. Arefin.

المنارة للاستشارات
www.manaraa.com

2.6.2. Quick select

In this section, we analyze the performance of the *k*-NNG construction with Quick-Select. Different tests were designed in order to showcase the performance of our algorithm with varying parameters, against *k*-NN algorithms presented in (Sismanis et al., 2012) and (Garcia et al., 2010). The Sismanis work uses a truncated bitonic sort for a *k*-NN search on GPUs. Finally, we benchmarked our code k_{th} element selection algorithm (Bexter, 2012). The k_{th} element algorithm selects the k_{th} largest/smallest values in a vector, and therefore is slightly different from the *k*-NN problem.

In Figure 24 the comparison of quick select *k*-NN search algorithm is shown with that of Garcia. A three-dimensional graph is chosen to represent the performance analysis of *k*-NN search algorithms with varying k and n. The data dimension is set to a constant value of d = 128 for all tests. The number of objects varies from n = 1024 to n = 131072, and the number of extracted nearest neighbors is varied from k = 8 to k = 512. As the figure shows, the speedup grows exponentially with increasing n. For small k and n, the speedup is 3×. For small n values, increasing k leads to 250× speedup. This speedup grows to up to 450× for large values of n with increasing k.

Figure 25 shows the performance advantage of our algorithm when benchmarked against truncated bitonic sort by Sismanis. Our speedup ranges from 1.5x for $k = 2^3$; $n = 2^{17}$ to 5.3x for $k = 2^9$; $n = 2^{17}$. We could not go above $k = 2^9$ since the Sismanis implementation would crash. It is obvious that the speedup saturates for large n. This saturation point is further away as the size of k grows. This clearly shows that for k > 29, the speedup would grow even more.

www.manaraa.com

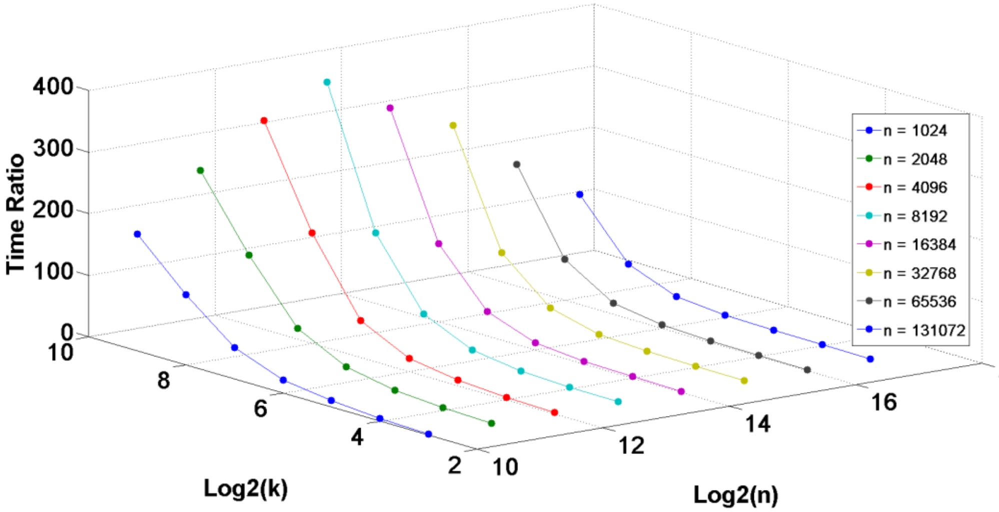


Figure 24: Quick-Select benchmarks against Garcia. In this set of tests the data dimension is set to be constant at d = 128 and k is doubled from k = 8 to k = 512. For each k, the performance graph is representing the timings for different n starting from n = 1024 to n = 131072.

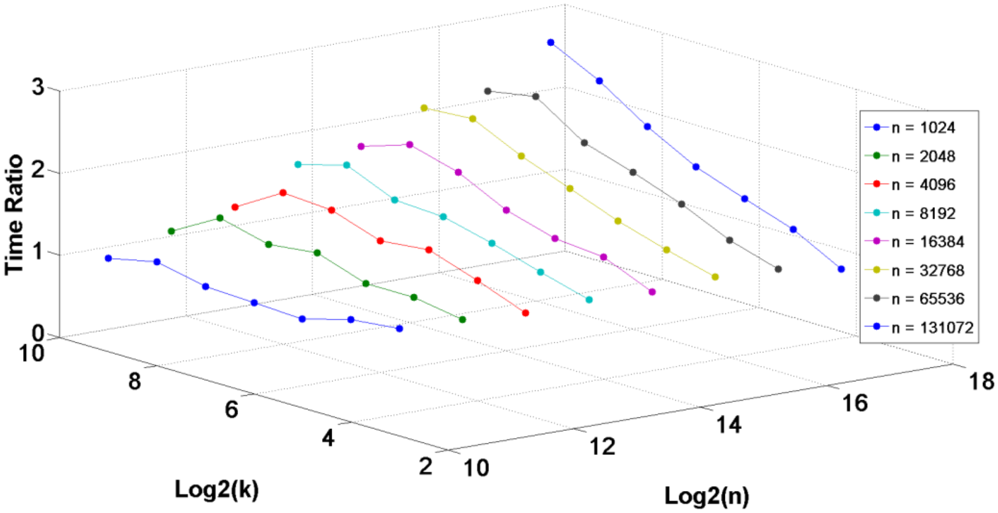


Figure 25: Quick-Select benchmarks against TBiS. In this set of tests the data dimension is set to be constant at d= 128 and k is doubled from k = 8 to k = 512. For each k, the performance graph is representing the timings for different n starting from n = 1024 to n = 131072.

Our benchmarks against the MGPU Select was for the selection algorithm alone. Note that the MGPU Select works for one query at time. Moreover, the MGPU Select

المنارة للاستشارات
www.manaraa.com

algorithm only selects the kth largest/smallest element. We conducted multiple queries by first loading the distance matrix in global memory on the GPU and then running the MGPU Select in successive rows, one row at a time. Furthermore, while our algorithm finds the k smallest elements with indices, the MGPU Select algorithm only finds the k_{th} smallest element. For finding the k smallest elements with indices, the MGPU Select algorithm is no better than a plain sort and selection (Bexter, 2012). Figure 26 shows the results. For small n, our algorithm has dramatic performance advantages (100×). This is because the GPU is not saturated by the MGPU Select. With increasing n, we see a significant drop off and possible saturation of the performance gain at around 8×. We are not able to explore a larger n because the distance matrix does not fit into GPU memory.

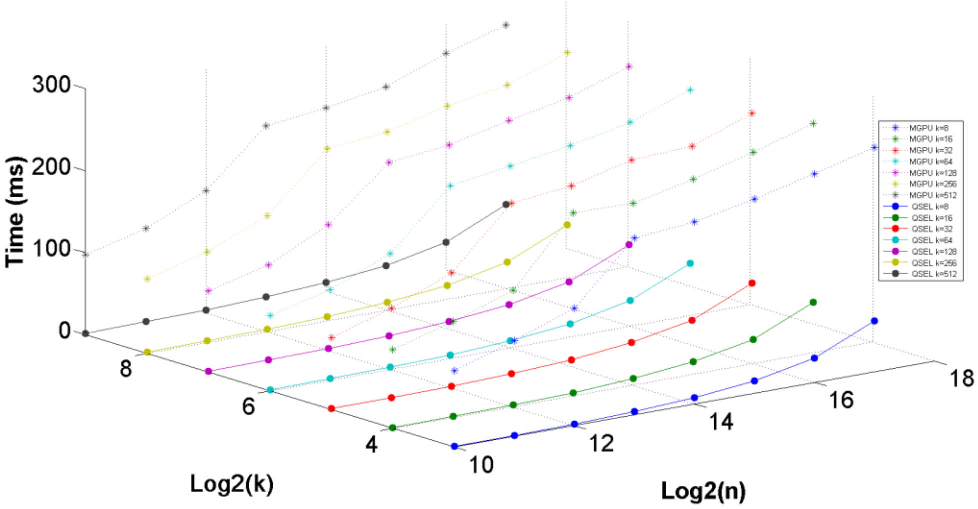


Figure 26: Timing benchmarks of quick select and MGPU select algorithms. In this set of tests k is doubled from k = 8 to k = 512. For each k, the performance graph is representing the timings for different n starting from n = 1024 to n = 131072.

www.manaraa.com

2.7. Conclusions

In this chapter, we presented a distributed GPU-accelerated implementation of the brute force k nearest neighbor graph construction method. Our implementation runs on an exclusive access GPU cluster. It forms a central core of a software pipeline for data and compute intensive Diffusion Map being used for structure and conformation recovery of biological entities from a large data set of noisy snapshots.

The contributions of this work are the following:

- A scheme for data partitioning and task assignment for efficient load-balanced execution of the brute force *k*-NNG method on a homogeneous cluster with GPU accelerated nodes. This implementation uses multiple levels of parallelism (between nodes, between multiple cores in nodes, and on GPUs) along with parallel I/O.
- Two new GPU algorithms for finding *k*-NNG from a given distance matrix

  -The first called batch index sorting uses three sort operations to directly find the *k*-NNG without further manipulation of the distance matrix.

  -The second is an efficient GPU implementation of the quick select algorithm and requires the computation of the transpose of the distance matrix for *k*-NNG construction.

  This implementation is the fastest method in its class with a nearly 4x gain over the state-of-the art.

www.manaraa.com

2.8. References

Adams, Bart, et al. "Adaptively sampled particle fluids." *ACM Transactions on Graphics (TOG)*. Vol. 26. No. 3. ACM, 2007.

Arefin, Ahmed Shamsul, et al. "GPU-FS-kNN: A Software Tool for Fast and Scalable kNN Computation Using GPUs." *PloS one* 7.8 (2012): e44000.

Arya, Sunil, et al. "An optimal algorithm for approximate nearest neighbor searching." *Proceedings of the fifth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 1994.

Barrientos, Ricardo J., et al. "kNN query processing in metric spaces using GPUs." *Euro-Par 2011 Parallel Processing*. Springer Berlin Heidelberg, 2011. 380-392.

Bell, Nathan, and Jared Hoberock. "Thrust: A 2 6." *GPU Computing Gems Jade Edition* (2011): 359.

Bentley, Jon Louis. "K-d trees for semidynamic point sets." *Proceedings of the sixth annual symposium on Computational geometry*. ACM, 1990.

Bexter, Sean. MGPU select, 2012.

Chen, Jie, Haw-ren Fang, and Yousef Saad. "Fast approximate k NN graph construction for high dimensional data via recursive Lanczos bisection." *The Journal of Machine Learning Research* 10 (2009): 1989-2012.

Coifman, Ronald R., and Stéphane Lafon. "Diffusion Maps." *Applied and computational harmonic analysis* 21.1 (2006): 5-30.

Connor, Michael, and Piyush Kumar. "Fast construction of k-nearest neighbor graphs for point clouds." *Visualization and Computer Graphics, IEEE Transactions on* 16.4 (2010): 599-608.

Cover, Thomas, and Peter Hart. "Nearest neighbor pattern classification." *Information Theory, IEEE Transactions on* 13.1 (1967): 21-27.

Dasgupta, Sanjoy, and Yoav Freund. "Random projection trees and low dimensional manifolds." *Proceedings of the 40th annual ACM symposium on Theory of computing*. ACM, 2008.

Datar, Mayur, et al. "Locality-sensitive hashing scheme based on p-stable distributions." *Proceedings of the twentieth annual symposium on Computational geometry*. ACM, 2004.

Dong, Wei, Charikar Moses, and Kai Li. "Efficient k-nearest neighbor graph construction for generic similarity measures." *Proceedings of the 20th international conference on World wide web*. ACM, 2011.

Dudoit, Sandrine, Jane Fridlyand, and Terence P. Speed. "Comparison of discrimination methods for the classification of tumors using gene expression data." *Journal of the American statistical association* 97.457 (2002): 77-87.

Garcia, Vincent, et al. "K-nearest neighbor search: Fast GPU-based implementations and application to high-dimensional feature matching." *Image Processing (ICIP), 2010 17th IEEE International Conference on*. IEEE, 2010.

www.manaraa.com

Haghani, Parisa, et al. "LSH At Large-Distributed KNN Search in High Dimensions." *WebDB*. 2008.

Indyk, Piotr. "Nearest neighbors in high-dimensional spaces." (2004).

Kato, Kimikazu, and Tikara Hosino. "Solving k-nearest neighbor problem on multiple graphics processors." *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*. IEEE Computer Society, 2010.

Khronos OpenCL Working Group. "The opencl specification." *A. Munshi, Ed* (2008).

Knuth, Donald E. *Art of Computer Programming, Volume 4, Fascicle 4, The: Generating All Trees--History of Combinatorial Generation*. Addison-Wesley Professional, 2006.

Kuang, Quansheng, and Lei Zhao. "A practical GPU based kNN algorithm." *International Symposium on Computer Science and Computational Technology (ISCSCT)*. 2009.

Levin, David. "Mesh-independent surface interpolation." *Geometric modeling for scientific visualization*. Springer Berlin Heidelberg, 2004. 37-49.

Luebke, David, et al. "GPGPU: general-purpose computation on graphics hardware." *Proceedings of the 2006 ACM/IEEE conference on Supercomputing*. ACM, 2006.

Moon, Bongki, et al. "Analysis of the clustering properties of the Hilbert space-filling curve." *Knowledge and Data Engineering, IEEE Transactions on* 13.1 (2001): 124-141.

NVIDIA, C. U. D. A. "Programming guide." (2008).

Pacheco, Peter S. *Parallel programming with MPI*. Morgan Kaufmann, 1997.

Paredes, Rodrigo, et al. "Practical construction of k-nearest neighbor graphs in metric spaces." *Experimental Algorithms*. Springer Berlin Heidelberg, 2006. 85-97.

Safar, Maytham. "K nearest neighbor search in navigation systems." *Mobile Information Systems* 1.3 (2005): 207-224.

Sanders, Jason, and Edward Kandrot. *CUDA by example: an introduction to general-purpose GPU programming*. Addison-Wesley Professional, 2010.

Schwander, P., et al. "Mapping the conformations of biological assemblies." *New Journal of Physics* 12.3 (2010): 035007.

Sismanis, Nikos, Nikos Pitsianis, and Xiaobai Sun. "Parallel search of k-nearest neighbors with synchronous operations." *High Performance Extreme Computing (HPEC), 2012 IEEE Conference on*. IEEE, 2012.

Volkov, Vasily, and James W. Demmel. "Benchmarking GPUs to tune dense linear algebra." *Proceedings of the 2008 ACM/IEEE conference on Supercomputing*. IEEE Press, 2008.

Wang, Jing, et al. "Scalable k-NN graph construction for visual descriptors." *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012.

Woo, Mason, et al. *OpenGL programming guide: the official guide to learning OpenGL, version 1.2*. Addison-Wesley Longman Publishing Co., Inc., 1999.

المنارة للاستشارات

www.manaraa.com

Chapter 3

Applications in determining the structure and conformations of biological entities

www.manaraa.com

3. Applications in determining the structure and conformations of biological entities

3.1. Summary

This chapter describes applications of developed Diffusion Map algorithm in determining structure and discrete conformational changes of biological entities with a bright perspective in high-resolution studies of wide variety of biological entities, namely: viruses, large proteins and other macromolecules.

First, we present the capabilities of Diffusion Map for recovering the structure of symmetric objects to $1/100_{th}$ of diameter from diffraction snapshots, leading to 3D reconstruction to atomic resolution using Satellite Tobacco Necrosis Virus (STNV) as simulated model system. High resolution structure recovery is made possible using highly optimized GPU implementation of Diffusion Map embedding, considering the fact that computational costs of embedding grows with eighth power of resolution elements.

Next, we present Diffusion Map for determining discrete conformational changes of the enzyme Adenylate kinase (ADK) from very large datasets of up to 20 million snapshots, each with $\sim 10^4$ pixels. This exceeds by an order of magnitude the largest dataset previously analyzed.

المنارة للاستشارات

www.manaraa.com

3.2. High resolution structure of viruses from random diffraction snapshots

3.2.1. Background

X-ray Free Electron Lasers (XFEL) ultra-short pulses provide a solution of a century long problem of acquiring damage free diffraction snapshots from intense radiation. XFEL opens new windows for single particle determination of structure of proteins and viruses. It also has the potential to provide information regarding conformational changes of biological entities. However theoretical methodology and algorithmic tools need to be developed to extract orientation and conformation signals from ultralow-signal diffraction snapshots.

As it is discussed in more detail in first chapter, different Bayesian and non-Bayesian algorithms took the challenge of determining structure from XFEL snapshots.. However, those methods lack enough resolution for determining structure of large proteins and viruses with ~100k atoms. As viruses become a common subject of study for XFEL due to their high scattering rate, developing high-resolution structure recovery algorithms become a must.

In addition, object symmetry is an important characteristic of viruses and many biological entities that has not been fully exploited by discussed algorithms. According to Shannon-Nyquist sampling theorem and in the presence of symmetry, reconstruction resolution r has a relation with number of available diffraction snapshots N_{snap}., the object diameter D, and the number of elements N_G in the point group of a symmetric object (Moths and Ourmazd, 2011):

www.manaraa.com

$$r = \left(\frac{8p^{2}}{N_{G} N_{snap}} \right)^{1/3} D \quad .$$

As the above equation shows, in the presence of symmetry, higher resolution is achievable using the same number of available diffraction snapshots. In current XFEL experimental setup, although there are some examples of high-resolution structure recovery using scattering snapshots (Seibert et al., 2011), the lack of sufficient “useful” single-particle snapshots (small N_{snap}) is a limiting factor (Yoon et al., 2011). Hereby the exploitation of symmetry is a crucial compensating element for scarcity of single-particle snapshots in high-resolution reconstruction. Being that said, there is no evidence of exploiting object symmetry for high-resolution reconstruction methods from single-molecule diffraction with their ultralow signal to noise ratio. We address this issue by introducing Diffusion Map, as a non-Bayesian manifold approach for high-resolution structure determination in the presence of object symmetry. By incorporation of object symmetry into Diffusion Map embedding, structure determination with resolution to $1/100_{th}$ of object diameter is achieved.

3.2.2. Methods

Diffusion Map and symmetry

The main theoretical assumption of manifold approaches, including Diffusion Map states that the scattering provides a "mapping" from object orientation to a diffraction snapshot. In a more general sense, scattering maps the orientation manifold, resulted from all orientations in rotation space, to a topologically equivalent compact manifold in the

المنارة للاستشارات

www.manaraa.com

diffraction snapshots space. The collection of all possible orientations in 3D space spans an SO(3) manifold. The SO(3) is mapped by scattering to a manifold governed by "symmetric top" metric in snapshots space. "symmetric top" manifold is roughly sketched as a sphere shortened in the direction of the incident beam as a results of projection (Giannakis et al., 2010). The mathematical formulation for describing "symmetric top" manifold is Wigner D-functions (Hu, 1973). These functions formulate the elements of a (3x3) rotation matrix in snapshots space (Hosseinizadeh et al., 2013).

In order to determine the orientation of snapshots, manifold approaches associate orthogonal Eigenfunctions of known operators from snapshots manifold into Wigner D-functions. Diffusion Map does that by describing snapshots manifold in terms of the Eigenfunctions of the Laplace-Beltrami operator with respect to an unknown metric (Coifman et al., 2005). Scattering from symmetric objects bring more complexity to the association with Wigner D-functions. In the absence of object symmetry, manifold produced by scattering is described by a homogenous metric to a good approximation. This leads the direct association of Eigenfunctions of Laplace-Beltrami operator to Wigner D-functions. However, in the presence of object symmetry, symmetrized Eigenfunctions are needed. Symmetrized Eigenfunctions are obtained by superposition of Wigner D-functions after operation by the elements of the object point-group:

$$\hat{D}^{j}_{m',m}(\alpha) = \frac{1}{N_G}\sum_{R_i \in G} O_{R_i} D^{j}_{m',m}(\alpha) \quad ,$$

www.manaraa.com

where a denotes the three numbers collectively representing any rotation, N_G the number of operations O_{R_i} in the point-group G, and $D^{j}_{m',m}(a)$ the (real) Wigner D-functions (Hosseinizadeh et al., 2013)..

For more detailed discussion about association of symmetrized Wigner D-functions please refer to our paper (Hosseinizadeh et al., 2013). Figure 27 compares symmetrized Wigner D-functions with snapshot manifold of an Icosahedral object by representing their spectra of Laplace-Beltrami operator.

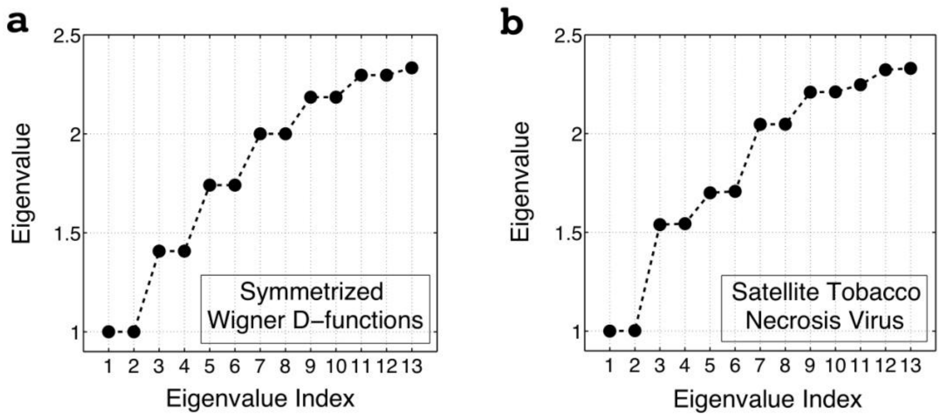


Figure 27: Comparison between Laplace-Beltrami operator spectra (Hosseinizadeh et al., 2013). (a) Spectrum for icosahedral Wigner D-functions. (b) Spectrum from the manifold produced by diffraction snapshots of the satellite tobacco necrosis virus.

Orientation recovery

Orientation recovery needs an association between Eigenfunctions of manifold produced by scattering snapshots and symmetrized Wigner D-functions. Having the association, orientation of each snapshot will be extracted from Wigner D-functions of low-dimensional mapped manifold produced by diffraction snapshots. Diffusion Map provides the governing dimensions of this manifold with respect to orthogonal empirical coordinates denoted here by Ψ_i . In principle, the task is to identify the association

المنارة للاستشارات
www.manaraa.com

between Ψ_i and symmetrized Wigner D-function. In the absence of substantial noise, eigenvalue spectrum comparison is enough for association. In the presence of noise and for more reliable association, scattering plots of embedded snapshots manifold in Ψ_i coordinates and symmetrized Wigner D-functions work as a perfect association mechanism. Figure 28 shows the comparisons of scattering plots. This association paves the way for recovering the orientation of each snapshot from its manifold in diffraction space (Hosseinizadeh et al., 2013)..

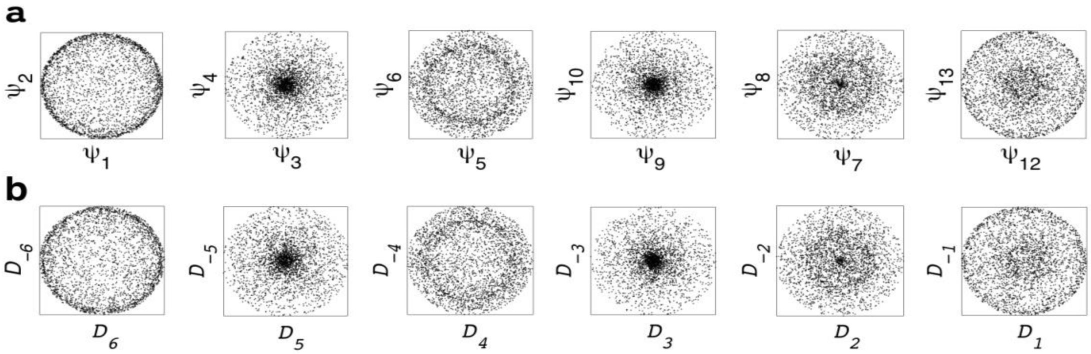


Figure 28: Association between pairs of (Ψ_i , Ψ_j) and Wigner D-functions (Hosseinizadeh et al., 2013).

Presence of symmetry introduces degeneracy between symmetrized Eigenfunctions. This adds an additional complication to the identification of the right Ψ_i . Since all orthogonal and normalized degenerate pairs of (Ψ_i , Ψ_j) is acceptable as a candidate for Wigner D-functions, any orthogonal operation, namely rotation, on the (Ψ_i , Ψ_j) pair is also providing an acceptable candidate. The following equation formulized the relation

المنارة للاستشارات
www.manaraa.com

between each degenerate (Ψ_i , Ψ_j) pair and its counterpart in Wigner D-functions via an unknown mixing angle, and a scaling factor (Hosseinizadeh et al., 2013):

$$\begin{pmatrix} \tilde{D}^{6}_{m} \\ \tilde{D}^{6}_{-m} \end{pmatrix} = \frac{1}{\sqrt{13}} \begin{pmatrix} \cos q_m & (-1)^{m+1}\sin q_m \\ (-1)^{m}\sin q_m & \cos q_m \end{pmatrix} \begin{pmatrix} y_i \\ y_j \end{pmatrix} .$$

In order to measure unknown mixing angle for each degenerate pair, we used mirroring over an axis to have conjugate snapshots in both diffraction space and more importantly in orientation space. For any snapshot, its conjugate, is simply its mirror image about an axis perpendicular to the incident beam axis passing through the center of the snapshot. (Figure 29).

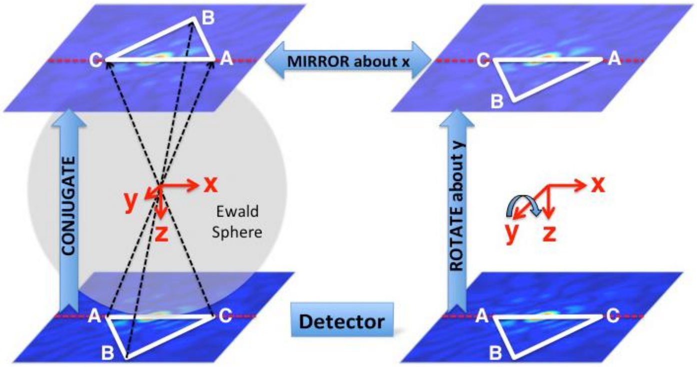


Figure 29: Schematic diagram of the mirroring process (Hosseinizadeh et al., 2013). The figure shows the relationship between the conjugate of an image, and its rotated version in the presence of Friedel symmetry.

Each snapshot and its conjugate are symmetrically located with respect to the axis of mirroring (Figure 30). Specifically, mirroring axis is perpendicular bisector of the line connecting an image to its conjugate. Since the conjugate of an image is easily obtained by a mirror operation through a line in the plane of the snapshots passing through its center, the mixing angle can be readily determined by adding the mirror images of a

www.manaraa.com

subset of the snapshots to the dataset before Diffusion Map embedding. Each mixing angle can then be determined to within π by the position of the perpendicular bisector of the lines connecting conjugate snapshots.

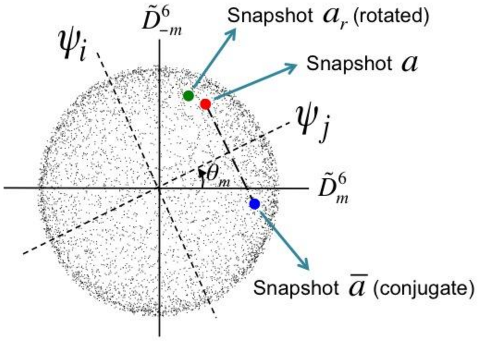


Figure 30: Finding the mixing angle, θm. By tracking the position of an snapshot with a few degrees rotation, the existence of an inversion in association with Wigner D-functions is found out.

Finally, there is a remaining π ambiguity between degenerate (Ψ_i , Ψ_j) pairs and their Wigner D-functions. This ambiguity stems from the chosen direction for the perpendicular bisector and is resolved by performing nonlinear square fits for each of the associated possibilities. The outcome of the fit, associated pairs with the lowest residual will eventually finalize the process of identifying association. The following cost functions measure the relation between the icosahedral Wigner *D*-functions and embedded Eigenfunctions for STNV model system in the presence of Icosahedral symmetry. For more information please refer (Hosseinizadeh et al., 2013):

www.manaraa.com

$$F_m = \begin{cases} \tilde{D}_0^6 - \frac{1}{\sqrt{13}}\psi_0 & \text{if } m = 0 \\ \tilde{D}_m^6 - \frac{1}{\sqrt{13}}\left(\psi_i \cos\theta_m + (-1)^{m+1}\psi_j \sin\theta_m\right) & \text{if } m > 0 \\ \tilde{D}_m^6 - \frac{1}{\sqrt{13}}\left((-1)^m \psi_i \sin\theta_m + \psi_j \cos\theta_m\right) & \text{if } m < 0 \end{cases} . \qquad (1)$$

This cost function is minimized using the “trust-region-reflective” nonlinear least-squares algorithm. The output is an estimated set of orientations (e.g., Euler angles) for each snapshot in the dataset, and a residual of the fit.

Simulating diffraction snapshots of the STNV

The STNV is used as a model system which presents Icosahedral symmetry. Molecular dynamics model of the virus, found in PDB under the name of 2BUK is used for generating diffraction pattern. The STNV has 20nm in diameter and 89,000 atoms. hydrogen atoms were neglected for simulation. X-ray diffraction patterns were simulated in random orientations to a crystallographic spatial resolution 0.2nm, having 12.4 keV photons. At first the 3D diffraction volume for all atom positions of the virus is calculated using Cromer-Mann atomic scattering factor (Cohan, 1958). Each diffraction pattern was obtained from the intersection of the diffraction volume with the Ewald sphere and subsequent projection onto a planar detector. The diffraction patterns were produced on a uniform grid of 443×443 = 196,249 detector pixels. Approximately 1.32 million diffraction snapshots are generated to give 0.2nm resolution for an object of 20nm. The orientations were sampled approximately uniformly as described by (Lavisolo and Da Silva, 2001).

المنارة للاستشارات

www.manaraa.com

GPU implementation

The procedure and pseud-ocode for embedding diffraction data with Diffusion Map are described in detail in chapter 2 and (Ginnakis et al., 2010). The proposed method is scalable with multi-levels of parallelism (between nodes of a cluster, between different GPUs on a single node, and within a GPU). The distance matrix calculation was performed on a parallel GPU cluster with 16 nodes, each equipped with two NVIDIA Tesla C2050 processors. The eigenvalue decomposition of the diffusion matrix was implemented with the Arnoldi factorization routine (ARPACK package; http://forge.scilab.org/index.php/p/arpack-ng/) using a single GPU for matrix-vector multiplication subroutines.

Given the characteristics of STNV diffraction snapshots dataset, extremely large dimensions and moderate batches, a modified task distribution between GPUs is implemented to exploit the full efficiency of GPU resources. Having the same assumption of 2 GPUs per node. Each node runs 3 CPU threads. 2 threads control the GPUs while one thread is in charge of I/O from the local disk as well as the shared disk.

Each node is responsible for computing a partition S(I,J) of the squared distance matrix. A_I s are read from the head node through parallel I/O. A_J s are read from the local disk. Within the node, A_I is divided into M=8 equal partitions. A_J is divided into R=8 partitions. which is governed by available GPU RAM. While reading A_I from memory is quite fast (it is parallelized), reading A_J from the local disk is slow. We therefore hide this latency by reading the disk in parallel with computation.

When the computation of $S_{(I,J)}(m, r)$ is complete, the column *k*-NNs as well as the row *k*-NNs are computed using sorting. However as opposed to general task distribution

www.manaraa.com

presented in chapter 2, $S_{(I,J)}(m, r)$ r = 1, 2,...,R are being sorted by the same GPU and at the same time. Due to small size of each $S_{(I,J)}(m, r)$, sorting them serially will under-utilizes the GPUs. Thereby by simultaneous processing of all $S_{(I,J)}(m, r)$ r = 1, 2,...,R in each GPU better efficiency and speed is achieved. Simultaneous processing of all $S_{(I,J)}(m, r)$ r = 1, 2,...,R needs proper data structure for indices of rows and columns. That will be achieved by changing the data structure of row and column indices to contain M blocks of data instead of one. As it is discussed in more detail in chapter two, our method of processing, batch index sorting, is insensitive to the ordering and structure of given data as long as the given index data structures matches the row and column indices of each element.

Three-dimensional reconstruction

After structure determination, finding orientations of all snapshots, the 3D diffraction volume is reconstructed by assembling the diffraction snapshots with the now found orientation. Structure determination by iterative phasing is most efficiently performed by Fast Fourier Transformation of a diffraction volume sampled on a uniform Cartesian grid, which requires interpolation. A so-called “Cone-Gridding” algorithm is used for that purpose, briefly described below.

To estimate the diffraction volume $A(q)$ at a point q, all diffraction patterns with scattering vector q are identified. Geometrically, the incident beam directions of these diffraction patterns form a cone, hence the name “Cone-Gridding”. To recover the real-space structure (electron density), the phases at each point in the 3D diffraction volume must be determined. Two iterative phasing algorithms are used in combination: Hybrid-

المنارة للاستشارات

Input-Output (HIO) (Fienup, 1978) and Charge-Flipping (Oszlányi and Suto, 2004). First, the HIO algorithm was applied for a few hundred iterations using a spherical support constraint with approximately the diameter of the virus. This did not produce good agreement between the (input) diffraction volume and the retrieved structure. Second, the output from HIO was used as input for the Charge-Flipping algorithm for refinement, with a threshold determined by the standard deviation of the input diffraction volume, and no support constraints. Charge-Flipping converged typically within a few thousand iterations (Hosseinizadeh et al., 2013).

3.2.3. Results

GPU performance

The STNV dataset presents a new challenge for Diffusion Map embedding with GPU cluster. having 0.154 M pixels for each diffraction snapshots, processed data dimensions for *k*-NN graph construction is an order of magnitude larger than tested datasets. Number of pixels in each snapshot has a direct relation with input file sizes and having large dimensions has its own challenges in the input I/O. However, because of expandability of design and with some small modifications, highly optimized implementation of *k*-NN graph construction and consequent embedding is obtained.

A 13-fold speedup is achieved in data processing as compared with an optimized method running on a cluster of CPUs for embedding dataset of 1.3 million simulated snapshots of STNV with 160k dimensionality. For the dataset in the present work comprising 1.3 M snapshots each with 0.154 M pixels, the distance calculation was performed in 36 hours.

www.manaraa.com

The execution time for the eigenvalue decomposition of the diffusion matrix or full dataset was about one hour.

Three-dimensional reconstruction

The structure of STNV as a model system icosahedral virus is reconstructed to 1/100th of its diameter. For the satellite tobacco necrosis virus (STNV, PDB designation: 2BUK) used here, this corresponds to atomic resolution (0.2 nm) (Figure 31). The ability of proposed approach to operate successfully at signal-to-noise ratios as low as ~10^{-2} has already been demonstrated with simulated and experimental datasets for a variety of systems. This is ample for dealing with a wide range of weakly scattering biological objects.

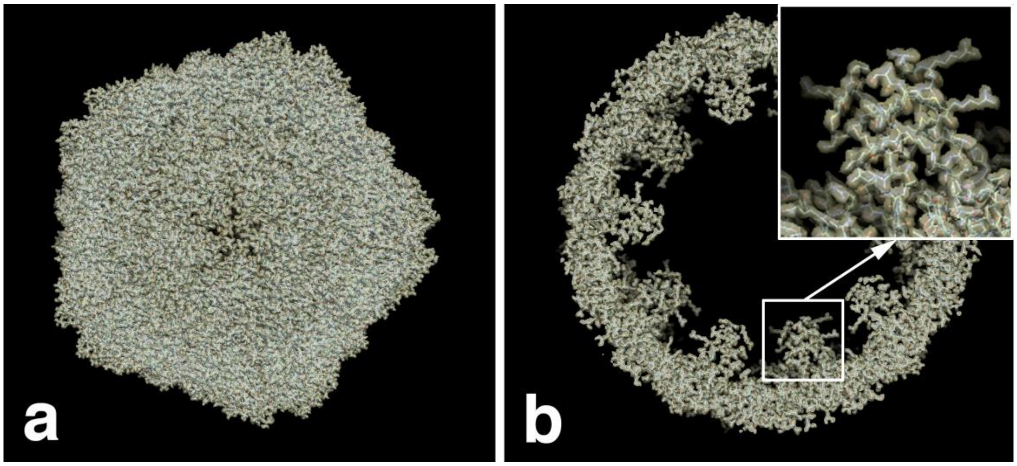


Figure 31: Atomic-resolution structure of the satellite tobacco necrosis virus (Hosseinizadeh et al., 2013). (a) The three-dimensional electron density map with resolution 1/100th of the object diameter (0.2nm). (b) A slice of the electron density showing atomic resolution.

For three-dimensional reconstruction, the snapshot orientations are recovered and used to compile a 3D diffraction volume. The electron density was obtained by iterative phasing

www.manaraa.com

(Oszlányi and Suto, 2004) to recover the atomic-level structure shown in Figure 31. Figure 32 compares cuts through the exact and recovered diffraction volumes. This completes the demonstration of structure recovery to 1/100th of the object diameter.

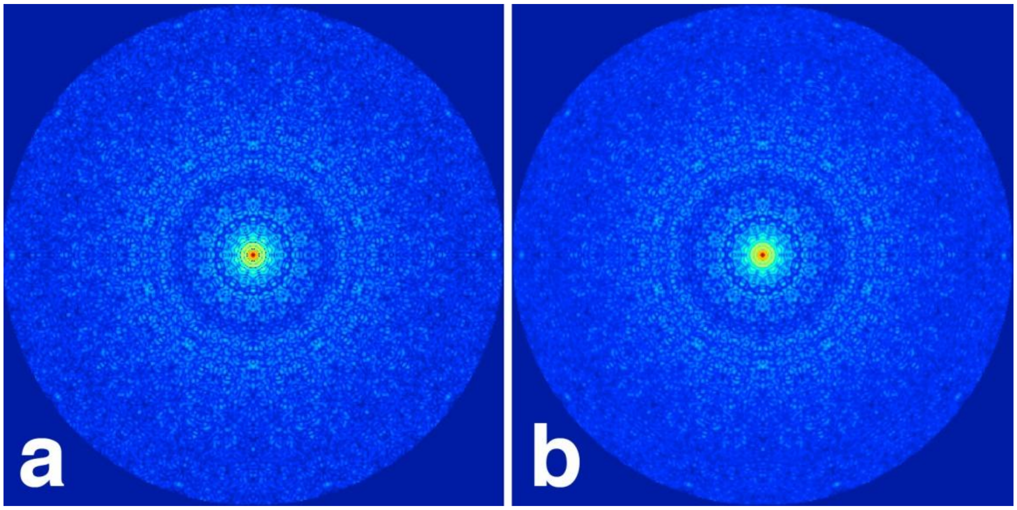


Figure 32: Comparison of the exact and recovered diffraction volumes (Hosseinizadeh et al., 2013). (a) The exact three-dimensional diffraction volume from simulate model system. (b) Same slice through the diffraction volume from recovered orientations.

3.3. Conformational heterogeneity in molecular degradation

3.3.1. Background

This section represents the power of manifold-based approaches for structure recovery alongside mapping conformational heterogeneity of biological entities passing the limits of radiation damage and ultralow signal to noise regimes.

Mapping conformational heterogeneity of biological entities is a topic of interest for x-ray crystallography and cryo-EM. Previous single particle reconstruction methods presume the absence of conformational changes in macromolecules and hence provided a limited knowledge of the conformations assumed by biological systems and their role in

www.manaraa.com

cellular function. Manifold-based embedding is proposed as a new approach to determine conformational changes alongside high resolution structure recovery of biological entities with existing and emerging experimental techniques (Schwander et al., 2010).

In general, manifold based algorithms, including Diffusion Map are not modality dependent and can be used with established experimental techniques such as X-ray crystallography and cryo-Em as well as emerging new techniques such as XFEL. It previously shown by our group that manifold-based methods are capable of determining structure of individual macromolecules from diffraction snapshots with high resolution (Schwander et al., 2012; Hosseinizadeh et al., 2013), and even in the very low radiation which is an order of magnitude below the previously required signals (Fung et al., 2008) (Figure 33).

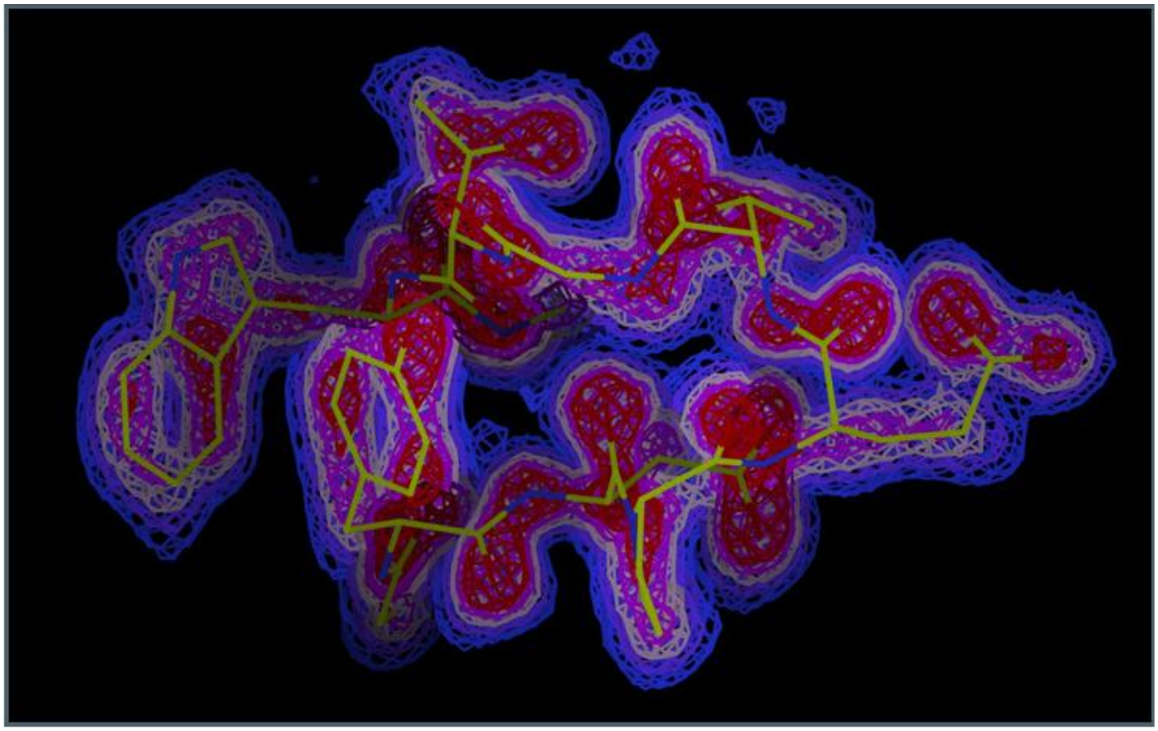


Figure 33: Chignolin reconstruction in the presence of noise (Fung et al., 2008). The structure is reconstructed from 72,000 simulated diffraction snapshots of the molecule at

المنارة للاستشارات

www.manaraa.com

random orientations and with mean photon count of 4×10^{-2} per pixel. Colors in the figure represents different atoms: C: yellow; N: blue; O: red.

The next step, which is the subject of this chapter, is to present the results of three-dimensional (3D) structure determination for different conformations of an object from random snapshots of an ensemble of *non-identical* objects, each in a different conformation This proposed application has significant impact on single particle imaging specially XFEL and cryo-EM. It also has the potential for inspiring future advanced tomographic studies of even more complex systems such as cellular organelles, Chromosomes and even the whole cell. Having a poor signal to noise ratio is one of the main challenges of mapping heterogeneity. Even in the absence of conformational changes, orientation recovery is challenging in ultralow SNR ensembles of diffraction snapshots and adding structural variability due to functional states of a molecule or a ligand-binding complicate the matter more.

Manifold based approaches is a new and powerful tool in structural biology to recover the structure and conformational changes of a biomolecule from a large ensemble of noisy snapshots obtained from unknown orientations and conformations. In the presence of different discrete conformations, manifold-based approach has the capability of unsupervised sorting of diffraction snapshots into different conformational classes and determine their orientation (Schwander et al., 2010).

Figure 34 shows the results of Isomap embedding of simulated diffraction patterns of randomly oriented over beam axis (one orientation degree of freedom) and randomly conformed into two conformations for the closed and open state of the molecule

المنارة للاستشارات

adenylate kinase (ADK; Protein Data Bank entries: 1ANK and 4AKE, respectively). Isomap is a nonlinear dimensionality reduction approach similar to Diffusion Map, that generates the similarity matrix based on geodesic distances between high-dimensional data point. The robustness power of manifold-based approach is demonstrated by embedding at a signal level corresponding to 4×10^{-2} photons pixel at 0.18 nm resolution (Schwander et al., 2010).

Due to their structural similarity and chemical identity, it is extremely hard to distinguish ADK conformational states using chemical methods. However, using Isomap embedding and with no prior information about the types and number of conformations and orientation degrees of freedom, all diffraction snapshots are sorted based on their conformational state and orientation. Considering the fact that the larger biological entities such as viruses and proteins scatter larger signals, it is therefore possible to reconstruct heterogeneous particles even with higher accuracy (Schwander et al., 2010).

المنارة للاستشارات

www.manaraa.com

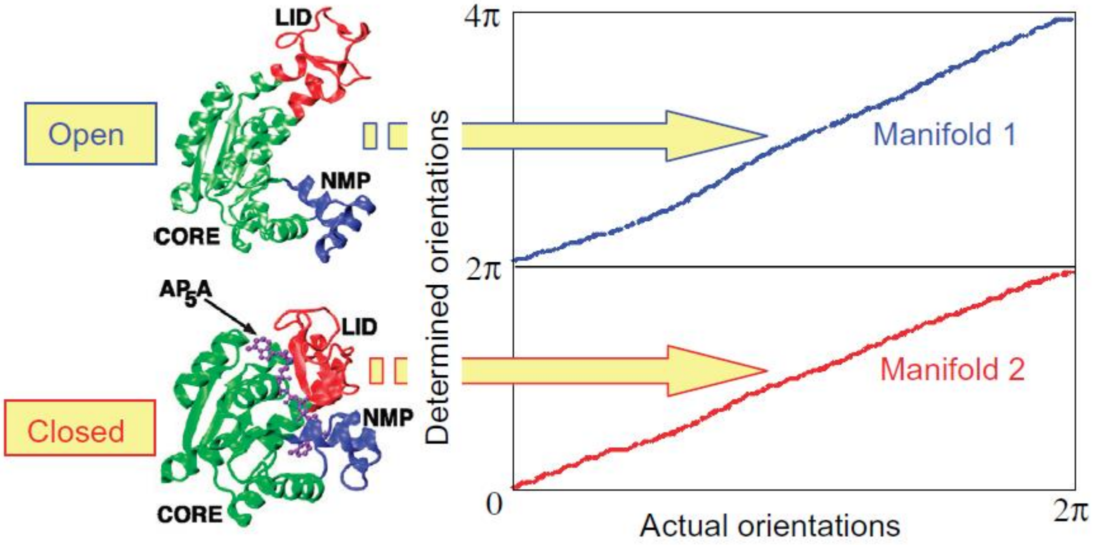


Figure 34: Mapping discrete conformations (Schwander et al., 2010). ADK molecule is reconstructed in two open and closed structures using simulated diffraction snapshots of molecules with no prior orientation and conformation information.

One ultimate goal of mapping heterogeneity is to find not only discrete conformational states of a molecule but also capability of mapping the entire continuum of conformations. Manifold-based approaches are shown to be successful at mapping and mapping the continuous changes of biological entities as well (Schwander et al., 2010).

In order to generate diffraction snapshots from a continuous molecular change. ADK in the process of melting is modeled by molecular dynamics. 12,500 diffraction snapshots were simulated from 100 conformations, with each conformation assuming 125 orientations about one axis. Figure 35 represents the results of Isomap embedding. As it shows the resulting three dominant Eigenfunctions enclose the manifold of conformational and orientation changes. It is clear from the image that structural heterogeneity and orientation variation over beam axis are combined and produced a tubular manifold. There are two directions of change in the manifold, the circular, closed

www.manaraa.com

cross-sections of the tube represents orientation signal and the paths along the main tube axis represent conformational change.

However the tubular manifold has a qualitatively obvious distinction between conformation and orientation directions, in general, orientation directions has a distinguishing element that separates them from directions corresponding to conformational changes in manifold. Considering the fact that orientation lies on SO(3) , more details in previous section, and has symmetries, the closed directions on the manifold that represents the SO(3) symmetry are for recovering structural orientations. SO(3) symmetry has been extensively established in general relativity and lattice space-time (Wald, 2010).

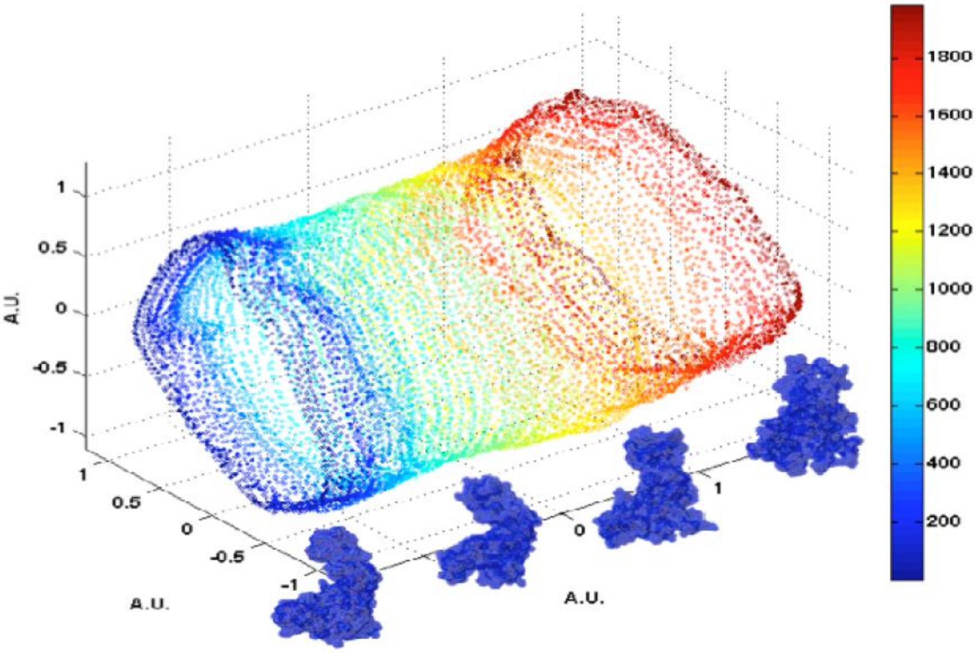


Figure 35: Embedding of continuous conformations (Schwander et al., 2010). Simulated diffraction snapshots of ADK model system is captured in 100 continuous conformations moving form closed to open conformation and randomly orientated along one rotation axis.

www.manaraa.com

Although efficient and successful at mapping structural heterogeneity, manifold-based approaches need a computational boost in order to be able to perform conformation and orientation recovery over the whole SO(3).

The computational costs of current algorithms rise rapidly with the number of resolution elements *r*. A rough calculation of computational complexity shows that it is in the order of $O(r^n)$ with $2 \leq n \leq 3$, where $r \approx (D/d)^3$, where *D* and *d* are the object diameter and resolution, respectively (Elser, 2009). Having c discrete conformational states will increase the computational costs by c^2 times as it increases the number of diffraction snapshots by c times. Generating this amount of diffraction snapshots is now experimentally possible by the advent of XFEL. The LCLS source and relevant detector operate at 120 Hz and are capable of delivering 10^7 diffraction snapshots per day.

Nevertheless there is another bottleneck in processing structural heterogeneity. As it is shown in (Schwander et al., 2012), the cost of embedding for structure recovery of fully SO(3) covered ADK snapshots in a single conformation is ≈56 hours using an exclusive CPU cluster with 100 MATLAB nodes. Using the same hardware platform for processing 10 conformations would take ~233 days and is physically impossible. Hereby and in an addition to our previous work, this section represents the capability of manifold-based approaches for unsupervised classification and 3D reconstruction of diffraction snapshots with unknown orientation and conformational changes using a GPU cluster to computationally boost up the processing and hereby bring it down to the realm of possibility.

www.manaraa.com

3.3.2. model system: melting ADK

In order to generate diffraction snapshots from a continues molecular change. ADK in the process of melting is modeled by molecular dynamics. The coordinates of ADK from *E. coli* in the open state (Protein Data Bank entry: 4AKE) were placed in a spherical droplet of water and simulated for 5 ns at a nominal temperature of 850K using NAMD (Philips et al., 2005). 20,000,000 diffraction snapshots were simulated from 10 conformations, with each conformation assuming 2,000,000 orientations.

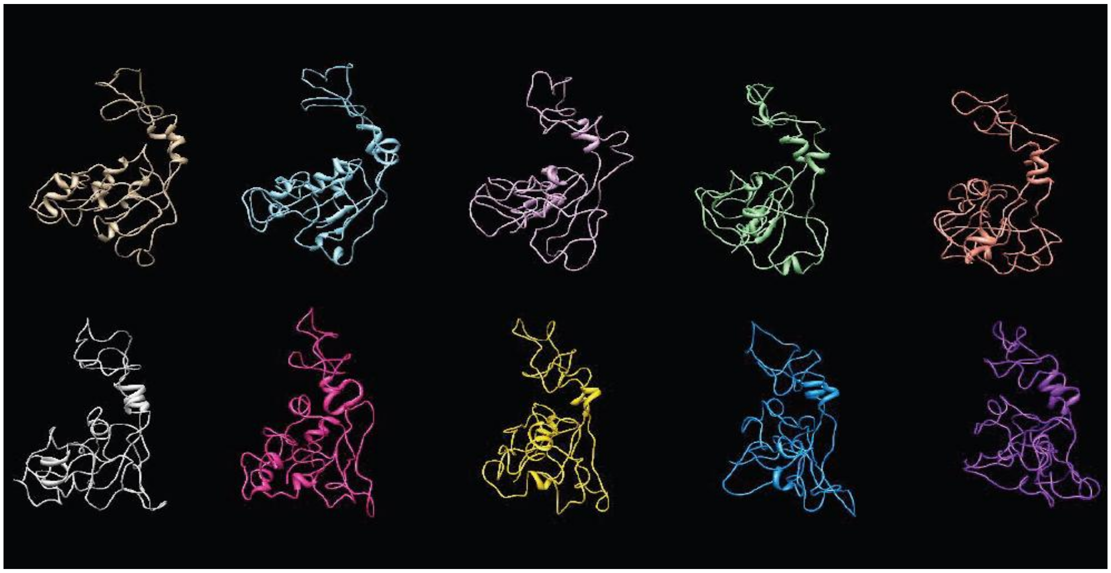


Figure 36: ADK model system in 10 different conformation.

X-ray diffraction snapshots of E. coli adenylate kinase (ADK; PDB descriptor 1ANK) in 10 different conformations from closed to open and in different orientations to a spatial resolution d = 2:45 Å, using 1 Å photons. Cromer-Mann atomic scattering factors were used for the 1656 non-hydrogen atoms (Cohan, 1958) of ADK in each conformation, neglecting the hydrogen atoms. The diffraction patterns were detected on a uniform grid

www.manaraa.com

of n = 126 × 126 = 15; 876 detector pixels of appropriate (Shannon) size, taking the corresponding orientations uniformly on SO(3) according to the algorithm in (Lavisolo and Da Silva, 2001). The number of diffraction patterns required to sample SO(3) adequately is given by $8\pi^2\ (D/d)^3 \approx 8.5 \times 10^5$, where D = 54 Å is the diameter of the molecule (Philips Jr., 1990) and d is the number of resolution elements. In our simulation, however, a larger D = 72 Å diameter was assumed, allowing, e.g., for the possibility to reconstruct the structure of ADK in different spectra of conformations. Hence, a total of s = 20×10^6 patterns were used in the present analysis.

As the below formula shows, the computational cost C of orientation recovery for each conformation in the absence of orientations scales as a power law:

$$C \approx O(R^8) = (D/d)^8$$

with D the object diameter and d the spatial resolution. The analysis of ADK was performed with R = 30 (R is the number of spatial resolution elements), compared with the largest previously published value of $R \leq 8$. This represents an increase of four orders of magnitude in computational complexity over the state of the art (Chapman et al., 2008).

3.3.3. Results

Performance benchmarks

To evaluate and apply our algorithms in manifold embedding, we executed *k*-NNG with batch index sorting for two data sets. The first data set contained two million images of simulated diffraction patterns of a randomly oriented adenylate kinase (ADK) molecule. Each image has 126 × 126 = 15876 pixels; i.e., high dimensionality. The second dataset

www.manaraa.com

consisted of twenty million images of simulated diffraction patterns of denaturing ADK in ten different molecular conformations. (For more information about the structure of data sets, please refer to (Schwander et al., 2012)). We evaluated the *k*-NNG algorithm with a previous implementation of a neighborhood graph construction by using MATLAB technical computing language. The MATLAB implementation took 56 hours on an exclusive CPU cluster with 32 nodes for two million diffraction patterns with the use of a highly optimized ATLAS-BLAS library for multi-threaded Matrix-Matrix Multiplication in double precision. The cluster had one Xeon E5420 quad-core CPU per node with 16kB of L1 cache, 6144kB of L2 cache and 40GFLOPS of double precision computing power. Since the parallel MATLAB implementation did not take advantage of the symmetry of the distance matrix, one can assume that such an implementation would take about 28 hours. Our GPU cluster had 16 nodes with each node equipped with two NVIDIA Tesla C2050 GPUs. Each Tesla C2050 GPU has a RAM of 3GB with 506GFLOPS of double precision computing power. There are 14 multi-processors sharing 720kB of L2 cache and each multi-processor having 48kB of user-configurable L1 cache/shared memory. In addition, each of the GPU nodes had two quad-core Xeon E5620s. Note that in our GPU cluster, the CPUs are used mostly for managing the GPUs and moving data between nodes and not for computation. Our GPU cluster implementation took 4.23 hours, giving a roughly 6.6× gain in performance.

To investigate the efficiency of our implementation we also benchmarked the most expensive part of the computation, i.e., matrix-matrix multiplication. We tested both double and single precision matrix-matrix multiplication on a single GPU (Tesla C2050) vs. a single core of an Xeon E5420 and concluded that if all four cores of the CPU were

المنارة للاستشارات

www.manaraa.com

active, we could achieve a roughly 7.7× speedup using GPUs just for matrix multiplication alone. As shown earlier, our complete implementation is slightly worse at a 6.6× gain in performance.

Based on the complexity of the Diffusion Map, the estimated execution time for ADK melting dataset with twenty million snapshots on the CPU cluster was more than eight months. With the use of Diffusion Map implementation on our GPU cluster, the execution time for twenty million snapshots was achieved in less than two weeks.

Table 1: Computational clusters and Diffusion Map total timings for simulated datasets.

	CPU #	GPU #	CPU	CPU RAM	GPU	GPU Global Memory	RME Total Time First dataset (Double precision)	RME Total time Second dataset
GPU Cluster	16	32	Intel(R) Xeon(R) E5620 2.4 GHz	48 GB	NVIDIA Tesla C2050	2.5 GB	5:23' Hrs	16 Days
CPU Cluster	32	0	Intel(R) Xeon(R) E5620 2.4 GHz	48GB	−	−	56 Hrs	Approx. 233 days

Mapping discrete conformational changes

Figure 37 shows the results of Diffusion Map embedding of 20,000,000 diffraction snapshots. It represents the scatter plot of first two dominant Eigenfunctions for all snapshots. As it shows, perfect clustering of ten conformations are achieved (different continents in the plot each representing 2,000,000 conformations).

المنارة للاستشارات
www.manaraa.com

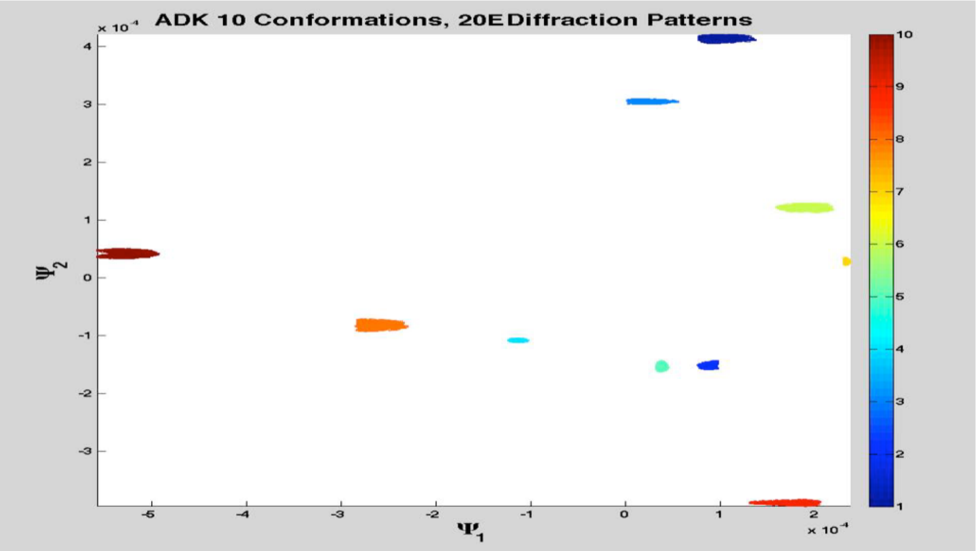


Figure 37: Diffusion Map embedding of ADK. Scattering plot of 20M snapshots reveals perfect clustering of 10 discrete conformations of ADK undergoing the process of melting.

Three-dimensional reconstruction

Perfect clustering of ten conformations make it easy to recover structure and have a 3D reconstruction for each of the conformations. A separate Diffusion Map embedding is used for diffraction snapshots of ADK molecule with closed conformation. In the second embedding, the purpose is to recover the orientation. The diffraction patterns are provided with no orientation information and orientation of each diffraction pattern is determined by method described in (Giannakis et al., 2010). The basic principle of structure recovery here is very similar to method proposed in previous section. Here, the goal is to provide rotation matrix for each snapshot based on identifying associations between Diffusion Map Eigenfunctions and Eigenfunctions describing SO(3) in the

المنارة للاستشارات
www.manaraa.com

diffraction snapshots space, symmetric top, which are taking the form of Wigner D-functions.

Next, after orientation recovery, diffraction volume is built by placing the diffracted intensities onto a uniform 3D Cartesian grid, as described by "cone-gridding", and deduced the 3D electron density by iterative phasing with a variant of the charge flipping technique. Reconstructed 3D molecule is in close agreement with the correct structural model is shown in Figure 38.

www.manaraa.com

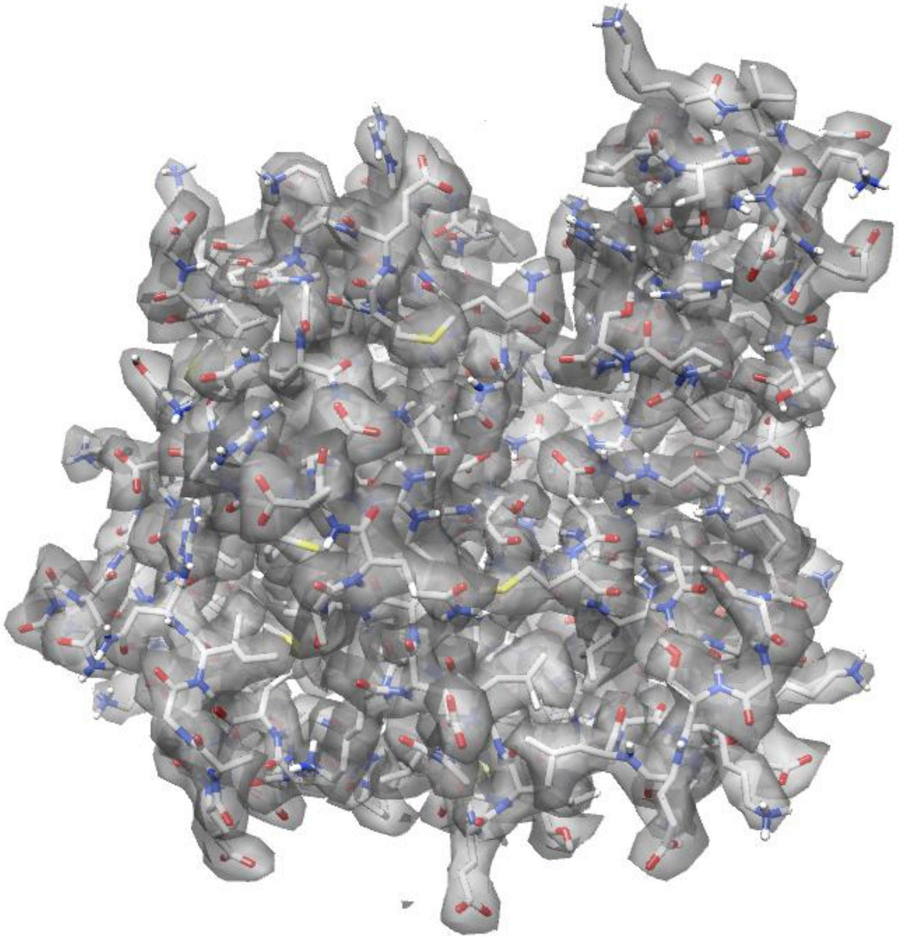


Figure 38: ADK in closed conformation (Schwander et al., 2012). Structure is determined using 2,000,000 randomly oriented diffraction snapshots at 2.45 Å resolution. Hydrogen atoms are neglected in electron density calculation demonstrated here.

3.4. Conclusions

In section 3-2, the first approach capable of extracting high-resolution 3D structure from diffraction snapshots of symmetric objects with structure recovery to 1/100th of the object diameter is presented. The proposed approach offers the possibility of applying manifold-

المنارة للاستشارات

www.manaraa.com

based techniques to study the structure of large biological molecules and viruses in atomic resolutions. Together with its previously demonstrated capability to operate at exceptionally low signal-to-noise ratios, Diffusion Map embedding has a bright perspective in processing XFEL diffraction snapshots of biological entities.

Section 3-3 represents the Diffusion Map embedding of 20 million simulated diffraction snapshots of Adenylate kinase (ADK) molecule in ten discrete conformations, with 2M snapshots each, capturing the process of melting. By the advent of XFEL single particle imaging techniques, number of diffraction snapshots per equipment run is increasing and hence having an expandable set of tools to overcome the computational expenses of embedding for growing data sizes becomes a necessity. This chapter presents developed Diffusion Map algorithm, as a powerful and scalable software tool to address the processing demands of growing XFEL datasets.

.

www.manaraa.com

3.5. References

Chapman, Henry N., et al. "Femtosecond time-delay X-ray holography." *Nature* 448.7154 (2007): 676-679.

Cohan, Norah V. "The spherical harmonics with the symmetry of the icosahedral group." *Proc. Camb. Phil. Soc*. Vol. 54. 1958.

Coifman, Ronald R., et al. "Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion Maps." *Proceedings of the National Academy of Sciences of the United States of America* 102.21 (2005): 7426-7431.

Elser, Veit. "Reconstruction algorithm for single-particle diffraction imaging experiments." *Physical Review E* 80.2 (2009): 026705.

Fienup, James R. "Reconstruction of an object from the modulus of its Fourier transform." *Optics letters* 3.1 (1978): 27-29.

Fung, Russell, et al. "Structure from fleeting illumination of faint spinning objects in flight." *Nature Physics* 5.1 (2008): 64-67.

Giannakis, Dimitrios, Peter Schwander, and Abbas Ourmazd. "The symmetries of image formation by scattering. I. Theoretical framework." *arXiv preprint arXiv:1009.5035* (2010).

Hosseinizadeh, A., et al. "High-Resolution Structure of Viruses from Random Snapshots." *arXiv preprint arXiv:1303.7253* (2013).

Hu, B. L. "Scalar waves in the Mixmaster universe. I. The Helmholtz equation in a fixed background." *Physical Review D* 8.4 (1973): 1048.

Lovisolo, L., and E. A. B. Da Silva. "Uniform distribution of points on a hyper-sphere with applications to vector bit-plane encoding." *IEE Proceedings-Vision, Image and Signal Processing* 148.3 (2001): 187-193.

Moths, Brian, and Abbas Ourmazd. "Bayesian algorithms for recovering structure from single-particle diffraction snapshots of unknown orientation: a comparison." *Acta Crystallographica Section A: Foundations of Crystallography* 67.5 (2011): 481-486.

Oszlányi, Gábor, and Andras Suto. "Ab initio structure solution by charge flipping." *Acta Crystallographica Section A: Foundations of Crystallography* 60.2 (2004): 134-141.

Phillips, James C., et al. "Scalable molecular dynamics with NAMD." *Journal of computational chemistry* 26.16 (2005): 1781-1802.

Phillips Jr, George N. "Comparison of the dynamics of myoglobin in different crystal forms." *Biophysical journal* 57.2 (1990): 381-383.

Seibert, M. Marvin, et al. "Single mimivirus particles intercepted and imaged with an X-ray laser." *Nature* 470.7332 (2011): 78-81.

Schwander, P., et al. "Mapping the conformations of biological assemblies." *New Journal of Physics* 12.3 (2010): 035007.

Schwander, Peter, et al. "The symmetries of image formation by scattering. II. Applications." *Optics Express* 20.12 (2012): 12827-12849.

Wald, Robert M. *General relativity*. University of Chicago press, 2010.

المنارة للاستشارات

Yoon, Chun Hong, et al. "Unsupervised classification of single-particle X-ray diffraction snapshots by spectral clustering." *Optics Express* 19.17 (2011): 16542-16549.

المنارة للاستشارات

www.manaraa.com

Chapter 4

Mapping conformational spectrum of the ribosome

www.manaraa.com

4. Mapping conformational spectrum of the ribosome

4.1. Summary

This chapter presents theoretical framework and developed algorithmic pipeline for mapping continuous conformational heterogeneity from a ribosome CryoEM experimental data. In this chapter, we build up on the Diffusion Map algorithm development and its applications in structure recovery and mapping discrete conformational changes from simulated data to capture the dynamics of ribosome from cryo-EM experimental data in the presence of overwhelming noise and in the absence of prior information about conformational changes. As a result we are presenting an approach capable of mapping ribosome continuous conformational changes over orientation space. Finally, cryo-EM projection snapshots of ribosome are connected via our analysis into iso-conformational, (only conformational changes, without orientation changes) and iso-orientational (changes of orientation for snapshots with same conformations) contours.

المنارة للاستشارات
www.manaraa.com

4.2. Background

Structural heterogeneity of ribosome has been the subject of extensive research in cryo-EM in recent years. A time-resolved detection of continuous conformational changes by extracting some loose timing information from experimental cryo-EM setup is presented by (Fischer et al., 2010). That revealed functional dynamics of ribosome along the motion of tRNA in both head and body of its subunits (More information in Chapter1). Beside this work and up to the author's knowledge, there is no algorithm capable of mapping continuous and time-driven conformational changes in ribosome from cryo-EM data.

In chapter 3, manifold-based approaches are represented with the ability of unsupervised detection of discrete or continuous changes in macromolecules in the presence of noise using simulated ADK diffraction snapshots. In next step, we tested the capability of Diffusion Map embedding to map conformational changes in simulated ribosome in the presence of ultralow signal. Using similar data preparation steps as ADK, a dataset of ribosome in two discrete conformational states, separated by attachment of a ligand, are prepared and 400,000 diffraction snapshots in random orientations uniformly distributed over orientation space are captured from each conformation. For simplification and better representation of results, in distance calculation step of Diffusion Map embedding, nearest neighbor diffraction snapshots were aligned by rotating along their set axis (with known rotation angle) before measuring their Euclidean distance. Hence diffraction snapshots lies on S2 manifold (sphere) in the new orientation space. As Figure 39 shows, three dominant Eigenfunctions of Diffusion Map presents a manifold consisting two

المنارة للاستشارات

www.manaraa.com

intertwined spheres similar to onion shells. This manifold recovers information regarding both conformational states and orientation of points for each conformation over S2.

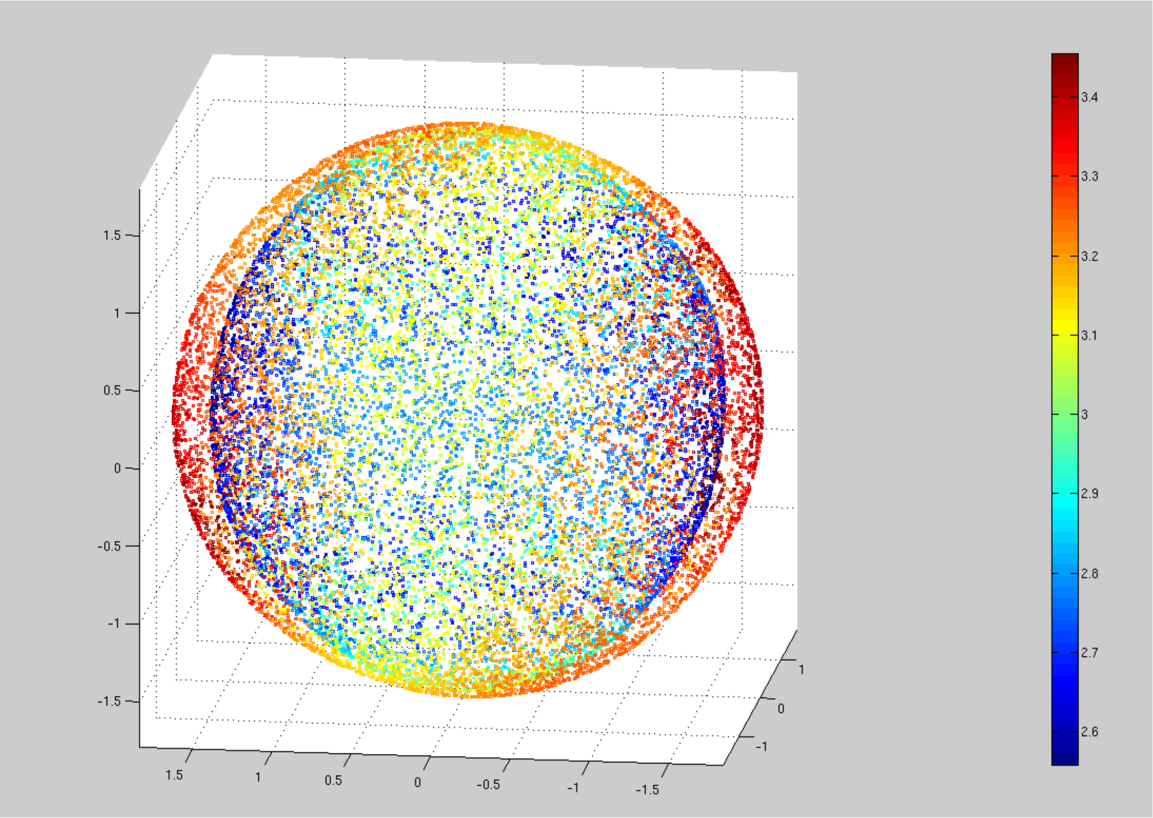


Figure 39: Manifold of ribosome with two discrete conformations (Schwander et al., 2013). Diffusion Map three dominant eigenfunctions recover both conformation and orientation of simulated ribosome in the presence of overwhelming noise.

Diffusion Map is used in next step for recovering conformational changes in experimental cryo-EM ribosome in the presence of overwhelming noise (around -12 db) and no prior information about types of conformational changes. Data is gathered by Y. Frank group in molecular biology department of Columbia university. The provided cryo-EM snapshots capture ribosome as a Brownian machine with spontaneous random motions that can be transferred into directed reactions by the presence of tRNA and

المنارة للاستشارات
www.manaraa.com

thermal energy fluctuations as it happens in the process of translation. Structure is recovered prior to our analysis by SPIDER toolkit for each snapshots.

SPIDER is an image processing software for reconstruction of biological molecules from the electron micrographs of single particles. It also provides a graphical user interface for better interactivity. Many people have contributed into development, improvement and maintenance of SPIDER over the past 30 years. (Shaikh et al., 2008) provides the most up-to-date information in this regards.

As it is discussed in more details in chapter 1, SPIDER uses two approaches, random-conical tilt and common lines to obtain an initial structure reconstruction of ribosome. Although manifold-based approaches are proven to be powerful tools for recovering orientation from diffraction snapshots and are even successfully used for orientation recovery of the cryo-EM projection snapshots, since orientation recovery in Cryo-EM data analysis seems to be a solved problem and SPIDER is used as a standard protocol for the purpose, it is chosen as primary software tool for recovering the orientations of snapshots.

Our initial analysis of experimental CryoEM data indicates that, without prior information about orientation of projection snapshots, Diffusion Map embedding of the whole dataset doesn't provide meaningful information regarding conformational states. Figure 40 demonstrates result of diffusion embedding with the same alignment of in plane rotation as it is discussed in the case of simulated data. As it shows dominant Eigenfunctions doesn't provide intertwined spheres. The resulting manifold is distorted in directions with more projection snapshots and do not represent any thickness, similar to onion shells in simulated data.

المنارة للاستشارات

www.manaraa.com

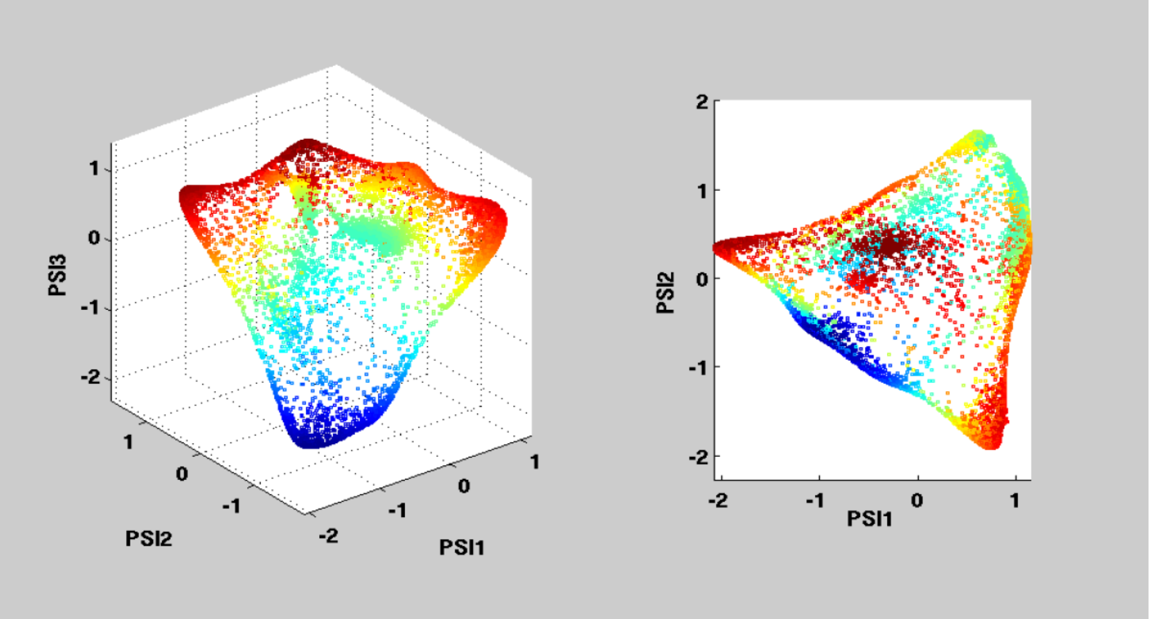


Figure 40: Diffusion Map embedding results of ribosome cryo-EM snapshots. As it shows, the structure of manifold is distorted by the anisotropy of distribution on orientation space and it does not reveal any clustering or parameterization of underlying conformational changes.

We believe there are two main reason for this lack of concordance between simulation and experimental data manifolds: First, it seems that the conformational changes are so tiny in comparison with orientation signal that are almost masked by its appearance in experimental manifold. This is not the case in discrete conformations of simulated data (Chapter 3 section 2) which presents a rather large conformation change, presence or absence of an elongation factor attached to ribosome. Secondly, the distribution of projection snapshots orientation over S2 sphere is heavily uneven (Figure 41), having three orders of magnitude difference in number of snapshots for highest and lowest populated regions of S2, with around two Shannon angle size. In contrast to experimental

www.manaraa.com

dataset, snapshots are uniformly distributed over S2 in simulated case, hence resulting in sphere-like shells in embedded manifolds.

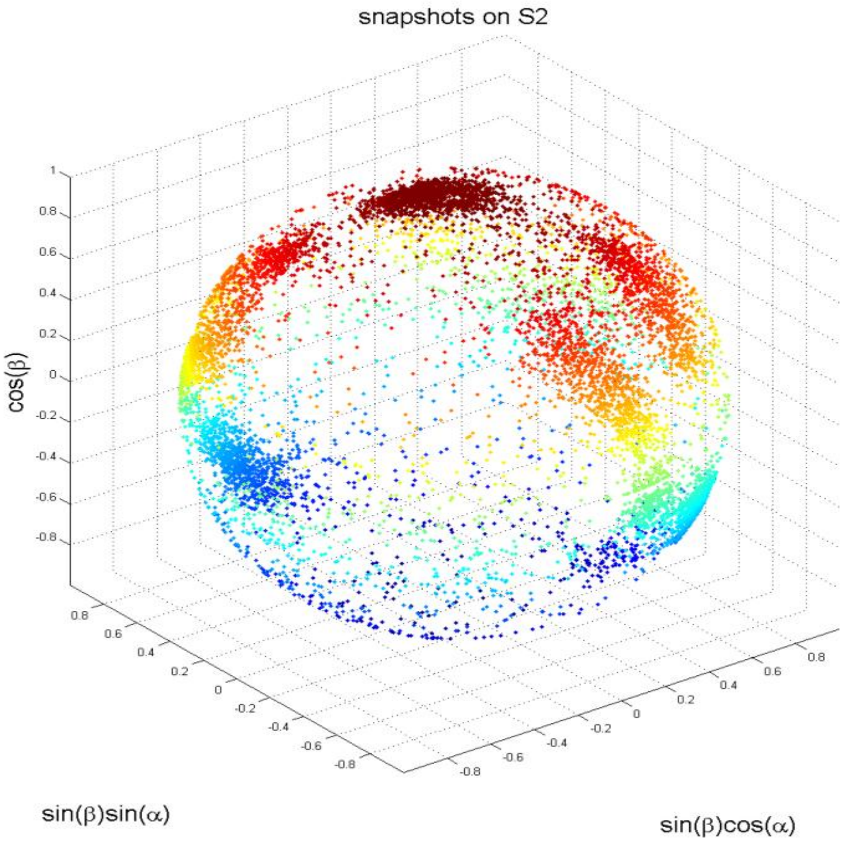


Figure 41: Distribution of snapshots on orientation space (S2). The distribution can be best described by discrete continents. S2 sphere is acquired from two Euler angles (α and β) disregarding the in-plane rotation angle, γ.

In order to capture the tiny conformational signal in this anisotropic dataset, a method is suggested to zoom in highly populated small (around two Shannon angle) tessellations of S2, called projection directions here on, and apply Diffusion Map embedding to capture local changes in each spot. By doing so, changes in orientation would be small enough

المنارة للاستشارات
www.manaraa.com

that would not have a significant effect in the resulting manifold. In addition and after analyzing snapshots in local directions, presumably in presence of conformational changes, those changes need to be connected to a global view in order to extract meaningful reconstructions of conformations.

These issues and more are organized in the following sections. In methods section, at first local embedding is discussed. The way we tessellate S2 into projection directions in each projection direction is presented in section 4-3-1. Next section (4-3-2) presents the methods we developed to enhance systematic variations of projection snapshots by CryoEM, namely the contrast transfer function correction. Third section deals with interpretation of Diffusion Map embedding results (4-3-3). Non-linear Laplacian Spectral Analysis (NLSA) is presented as a method of unwrapping information in one-dimensional manifold, which happens to be the case in our local embedding in 4-3-4. At last, we present our developed method for connecting the results of local embeddings into the global view in order to have a iso-conformation and iso-orientation contours of changes over S2 in section 4-3-5.

In the end, we present the results of initial embedding, correlation analysis of the observed manifold with systematic factors and iso-conformation and iso-orientation maps of data. At the end we complete this chapter by having a conclusion over the presented results.

المنارة للاستشارات

www.manaraa.com

4.3. Methods

4.3.1. Dataset characteristics

Cryo-EM ribosome dataset contains 849,919 projection snapshots in the presence of overwhelming noise (around -12 db). There is not any published information about types and discreteness of conformations in the dataset. Data is provided by Frank group in molecular biology department of Columbia university and captures ribosome as a Brownian machine with spontaneous random motions in the presence or absence of tRNA in the process of translation. Orientations of all snapshots is provided prior to our analysis by SPIDER toolkit for each snapshots.

4.3.2. Projection directions

Projection snapshots are tessellated over S2 based on their given rotation angles from SPIDER, omitting rotation along beam axis, using tessellation method proposed by (Lovisolo and Da Silva, 2001). The basic idea behind Lovisolo method is to distribute uniformly *K* points on an N-dimensional hyper-sphere. The main supposition of the proposed method is the following: having a large enough K, a set of vectors will be found that are uniformly distributed. That set define a tiling of equally length hyper-cubes over the hyper-surface. The proposed algorithm starts with a set of K' vectors (K' > K) with identical tiling that cover the whole hyper-surface and iteratively decrease K' without losing the covered area until convergence at K' =K. In our application, we chose K=1500 which is the minimum number of snapshots uniformly distributed over S2 that can give back the three-dimensional reconstruction with the required resolution. After finding the

www.manaraa.com

tiling over S2, projection snapshots are classified into each tessellation based on their position over S2. Figure 42 presents the results of Lovisolo tessellations.

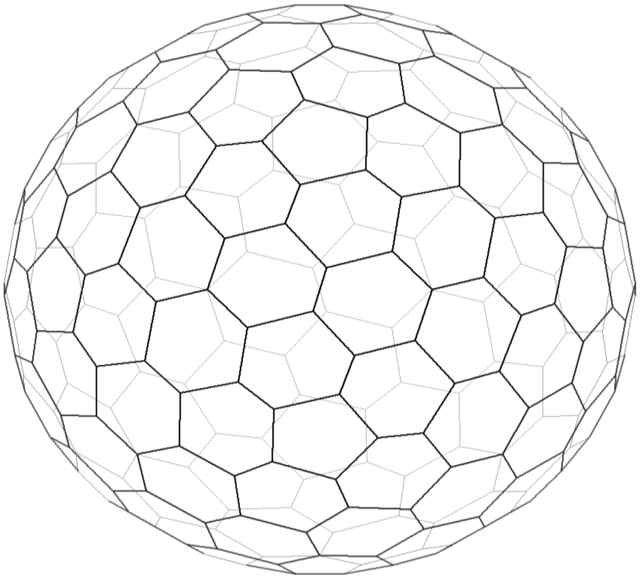


Figure 42: Schematic view of Lovisolo tessellations of a sphere.

After tessellating S2 and classification of snapshots into projection direction, we sort projection directions based on their population to give a quantitative measure of Anisotropy of dataset. Figure 43 presents sorted population of all 1500 projection directions. As it shows, there is a severe anisotropy in number of projection snapshots for each projection direction. The population ratio between highest and lowest populated areas is 1028.

www.manaraa.com

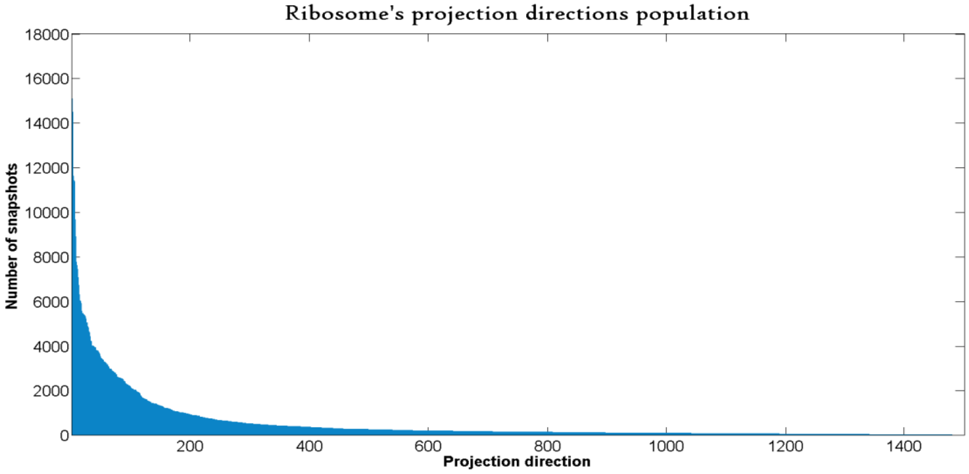


Figure 43: Anisotropy of Cryo-EM dataset. Sorted population of all projection directions shows a severe (3 orders of magnitude) anisotrpy in population of projection directions.

4.3.3. Contrast transfer function of CryoEM micrographs

Contrast transfer function is a measure of true resolution for an electron microscope. As it is discussed in more details in Chapter 1, in cryo-EM macromolecules and biological entities of interest are frozen in batches on a thin layer of ice and their scattering electron density make a trace on a detector. Each micrograph can be approximated as modulation of a plane in Fourier space by a function that represents the resolution of electron microscope. This function is called CTF and is formulized as bellows:

$$CTF(s) = -k(\sqrt{1 - Q^2}\sin\gamma + Q\cos\gamma\ .$$

Where we have:

$$\gamma = -2\pi(\frac{C_s\lambda^3 s^4}{4} + \frac{(z_0 + z)\lambda_s{}^2}{2})$$

المنارة للاستشارات

Q is fractional amplitude contrast coefficient, s is Fourier 2D frequency amplitude, λ is electron wavelength, C_s is spherical aberration coefficient, z_0 is the object lens defocus and z is relative vertical position with respect to z_0.

Having this formula and for each image radius in Fourier space (fixed s), CTF is roughly viewed as a sinusoidal function of object lens defocus, and depth of each ice layer of particles from the origin of electron beam. Similar to most current reconstruction methods, we have the fixed layer depth assumption ($z = 0$) and viewing CTF as a sinusoidal function of object lens defocus z_0. In cryo-EM datasets, for each projection snapshot there is a different CTF function, based on the defocus value of lens in the time of capture. Figure 44 represents the CTF functions for a random snapshot from a projection direction in our dataset:

المنارة للاستشارات

www.manaraa.com

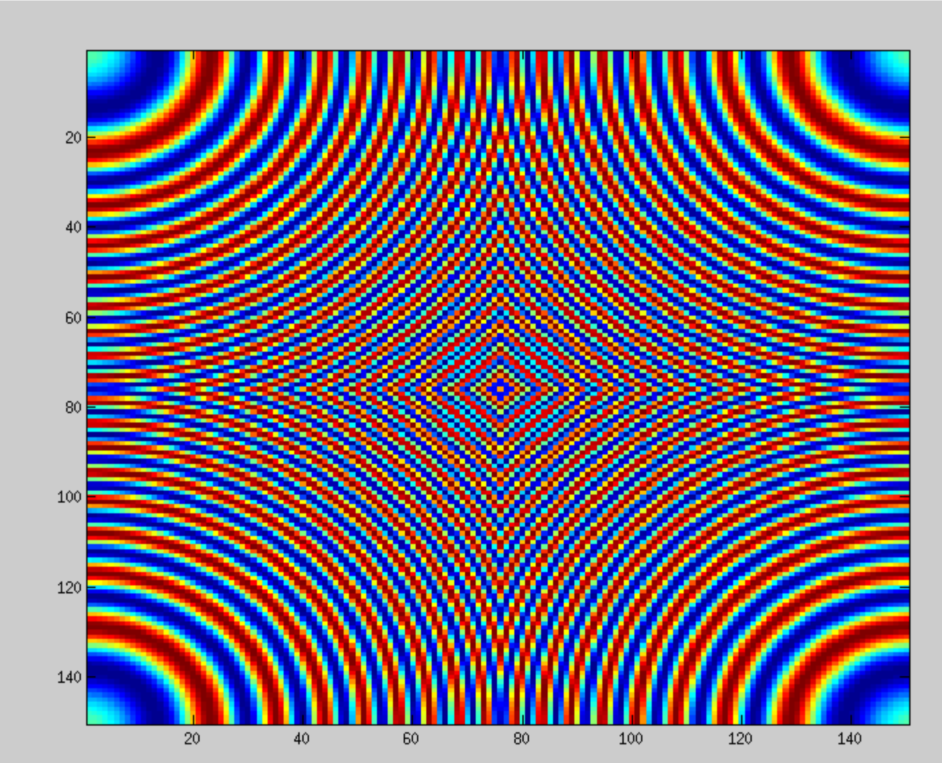


Figure 44: Contrast Transfer Function of a projection snapshot. CTF is a sinusoidal function of object lens defocus.

4.3.4. Local embedding

To have a correct measure of similarity between snapshots in each projection direction, two modifications are applied in Diffusion Map distance calculation routine in order to align snapshots with different rotation angles along the beam axis and correct the CTF difference between snapshots.

At first, all the snapshots of a projection direction are aligned along the beam axis with a reference snapshot. The alignment is applied with in plane rotation of images by difference rotation angle from reference snapshot. As it discussed in background section, all rotation angles are a priory provided by SPIDER software. The choice of reference image is arbitrary in each projection direction but to have representative alignment and

www.manaraa.com

hence snapshot average between all directions, projection snapshot in the center of each projection direction is chosen as the reference. After alignment by summing over snapshots, a noise reduced view of molecule in given projection direction is appearing. The average image is indeed the average of all conformations in a given projection direction and more powerful algorithms are needed for capturing the tiny differences due to conformational change.

Figure 45 represents the raw snapshot and the average of aligned snapshots. As it shows, due to overwhelming level of noise (~ -12 db) 2D view of the molecule is not recognizable by eye. However aligning the snapshots and averaging, works as a low pass filter and reduces noise to some extent.

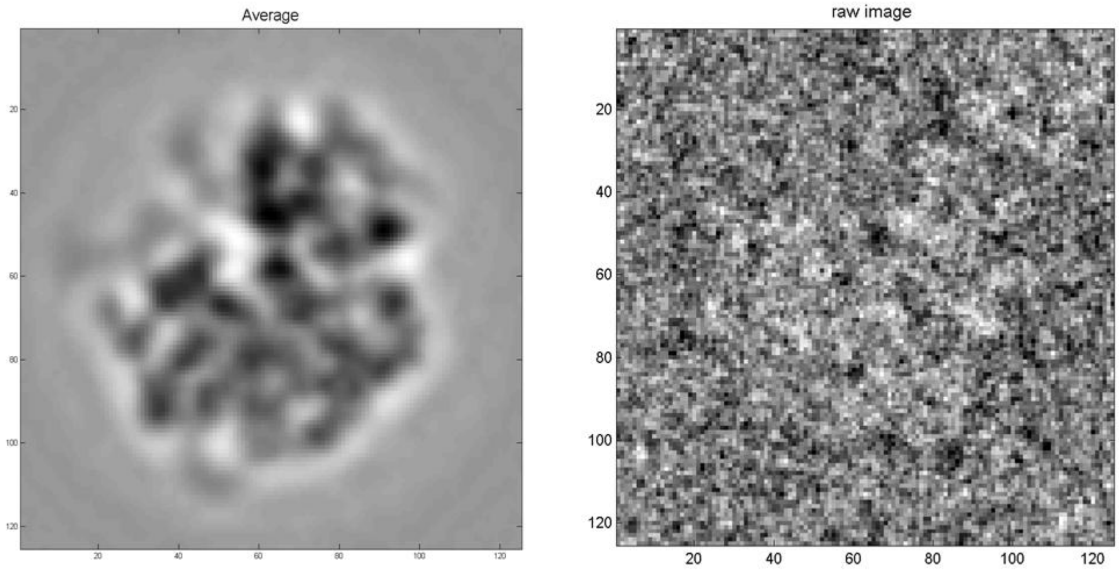


Figure 45 Raw (left) and average (right) 2D view of snapshots. Averaging made possible using alignment to a reference by in-plane rotation. 200 snapshots are averaged to get the right image.

Diffusion Map distance is a measure of sum of intensity difference of two snapshots. For snapshots with different CTF we have:

www.manaraa.com

$$I_1' = CTF1\ \times\ I_1$$

$$I_2' = CTF2\ \times\ I_2$$

Where I_1 and I_2 are the intensity snapshots in Fourier space that we are interested in finding their difference and I'_1 and I'_2 are the CTF filtered images from detector. In order to reduce the effect of CTF in difference density we use double filtering method proposed by P. Schwander. In this method and for calculating the difference intensity in Fourier space, intensities are double filtered by the opposite CTFs. New difference function has the following form:

$$Diff = \frac{(CTF2 \times I_1' -\ CTF1\ \times\ I_2')}{CTF1 \times CTF2} = (I_1 -\ I_2)$$

Using the following formula, the difference intensity information will be kept in every pixel except the ones in the zero crossings of both filters. Otherwise, summation of pixel intensity difference function would give a desired difference as a measure of distance for Diffusion Map.

4.3.5. Nonlinear Laplacian Spectral Analysis

Nonlinear Laplacian Spectral Analysis (NLSA) is proposed by (Giannakis and Majda, 2012 I) and further developed in (Giannakis and Majda, 2012 II) to capture nonlinear features of data generated by complex systems. NLSA has many applications in molecular dynamics (Lima et al., 2009; Hsieh, 2007), fluid dynamics (Dee et al., 2011) and even astrophysics (Christiansen, 2005; Vautard and Ghil, 1989). The goal of NLSA is to capture physically meaningful information regarding the spatio-temporal evolution of a given system from experimental data in order to have better understanding of

المنارة للاستشارات
www.manaraa.com

underlying dynamics and predictive capabilities (Giannakis and Majda, 2012 II). This method has been used in this chapter to further analyze the time and average evolution of one-dimensional manifold that happens to be the results of local embedding (Figure 46).

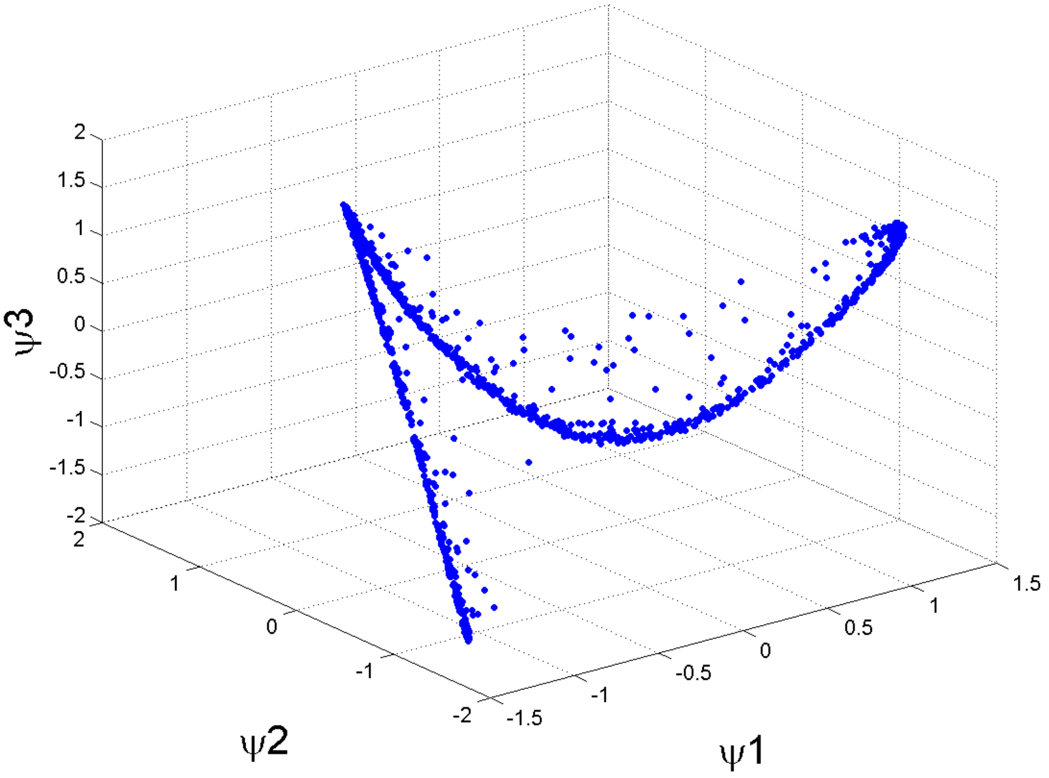


Figure 46: NLSA is used to interpret evolution of one-dimensional manifolds. Scatter plot of first three Eigenfunctions for all snapshots in one projection direction.

Similar to linear approaches such as singular spectrum analysis (SSA), NLSA uses spectral analysis of linear operators to decompose data into corresponding space of spatial patterns ("topos" space) and a space of temporal modes ("chronos" space). However and in contrast to linear methods, NLSA uses the manifold structure of data for decompositions. Having a one-dimensional manifold from Diffusion Map embedding of

www.manaraa.com

snapshots, NLSA gives Eigenimages and their nonlinear evolution as a result of walking through curvature of the manifold.

In order to apply NLSA on a one-dimensional manifold the following steps are taken:

1. **Ordering**: Data points on the manifold need to be ordered according to manifold direction. For this purpose, geodesic distance is used as a measure of distance between points on manifold. Method proposed by (Tenenbaum et al., 200) is used to calculate the geodesic distance between points on a manifold. Points on manifold are ordered by their geodesic distances to a corner point. Figure 47 demonstrates the Geodesic path between points in a graph and Figure 48 demonstrates the ordering step.

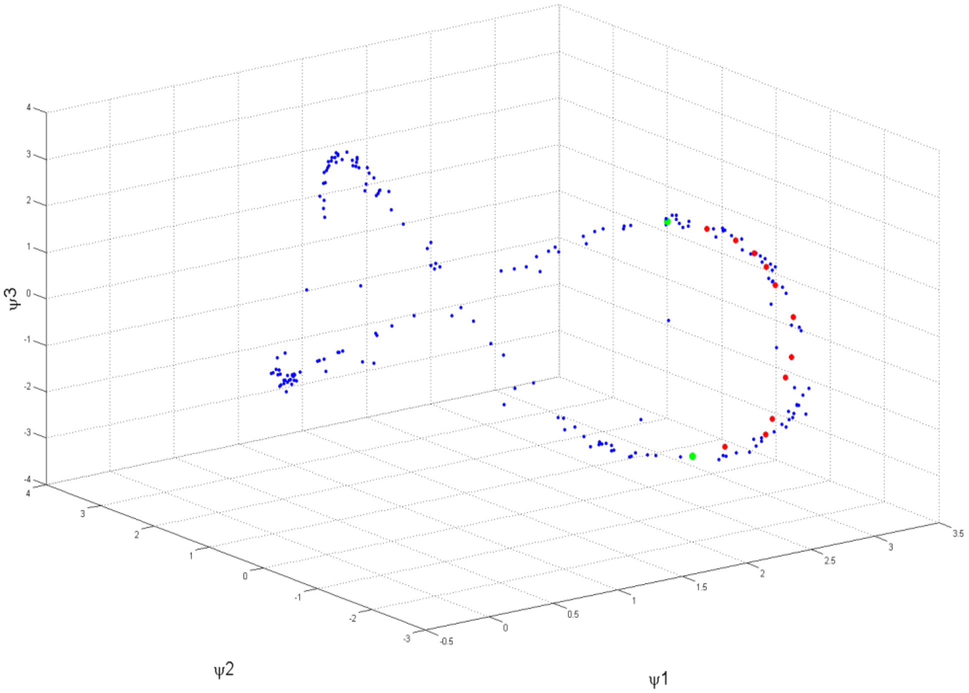


Figure 47: Geodesic distance as a measure of ordering along manifold. Geodesic path (in red) measured by geodesic distance between two points (in green). As it demonstrates Geodesic distance gives a measure of closeness based on graph properties of data.

المنارة للاستشارات
www.manaraa.com

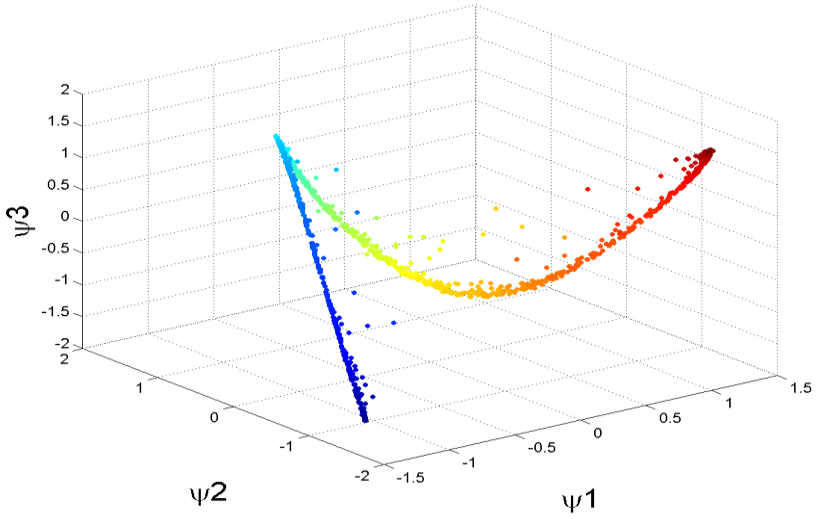


Figure 48: Geodesic ordering. Snapshot are ordered based on their geodesic distance to the point in lower left corner. Manifold is color-coded based on the geodesic distance values from small (Blue) to large (red).

2) **Concatenation**: After ordering points on manifold, their corresponding snapshots need to be concatenated. Concatenation provide the directionality information needed for temporal evolution of the system. Assuming a set of ordered projection snapshots X = $\{X_1,X_2,..X_n\}$ where X is a snapshot in a form of vector with d dimensions (d is number of intensity pixels) and having concatenation order C, concatenated set is:

$$X' = \{X_1^{'}, X_2^{'}, ..., X_n^{'}\}$$

where: X'_i is a C×d vector consisting of $\{X_i, X_{i+1}, ...X_{i+c-1}\}$.

3) **Embedding**: After concatenation, new set of vectors is used to embed with Diffusion Map to have a manifold structure for NLSA decomposition. Diffusion Map embedding is applied next on the concatenated set to provide the new manifold with the desired directionality being pronounced. Figure 49 is showing the results of embedding concatenated data.

www.manaraa.com

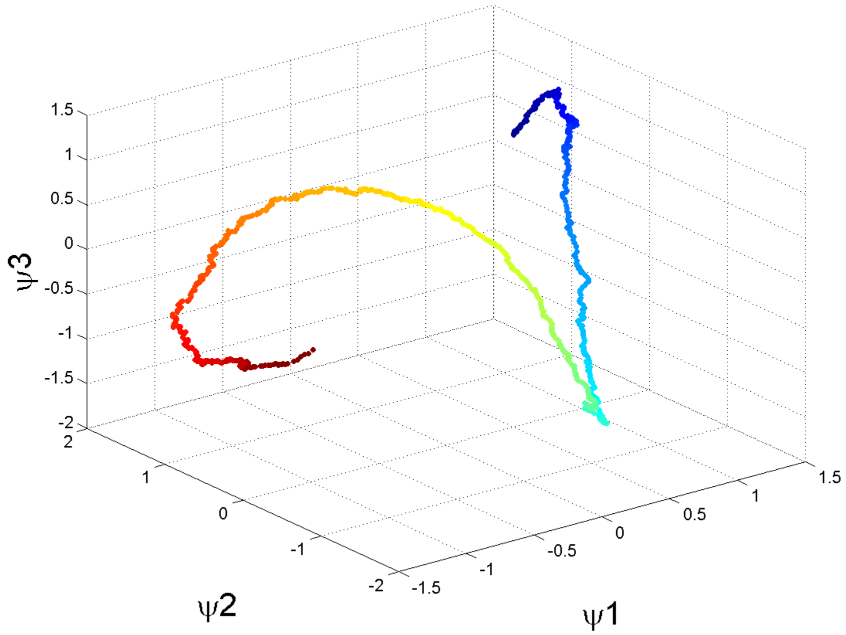


Figure 49: Diffusion Map embedding of concatenated data.

4) **Singular Value decomposition of A = X'μΦ**

From Diffusion Map embedding of concatenated data, Riemannian measure μ and projection manifold Φ are extracted. Riemannian measure is and n×n matrix containing the first trivial Eigenfunctions of Diffusion Map. Riemannian measure can be seen conceptually as the weight of embedding for each data points in reconstruction. Φ is n×L matrix of L retained Eigenfunctions of embedding and X' is the concatenated snapshots matrix. Matrix A can be seen as the projection of data onto the concatenated manifold.

By applying singular value decomposition (SVD) on projected matrix, spatial and temporal modes of change are decomposed from A, by the following formula:

$$A = USV^T$$

المنارة للاستشارات

Where U is the matrix containing the "topos modes" for spatial dynamics and V entails "chronos modes" for time evolution characteristics on the manifold. S is n×n diagonal matrix containing the singular values of SVD. Singular values are a measure of power for each mode of SVD.

5)**Back projection:** Although V entails temporal evolution but the time information and ordering are from projected data on the embedded manifold. In order to have temporal information on data space, V needs to be projected back into data space. This has been done by the following equation:

$$V^T = V^T \Phi^T$$

5) Reconstruction: Having the U, S and back-projected V matrices, data can be reconstructed using up to L topos and chronos modes:

$X'_k = U_k \times S_{kk} \times V_k$

Where X'_k is the k_{th} mode of X' matrix (with concatenated snapshots) reconstructed using k_{th} mode of topos and chronos. Figure 50 shows the first modes of topos and chronos for a reconstructed data:

المنارة للاستشارات
www.manaraa.com

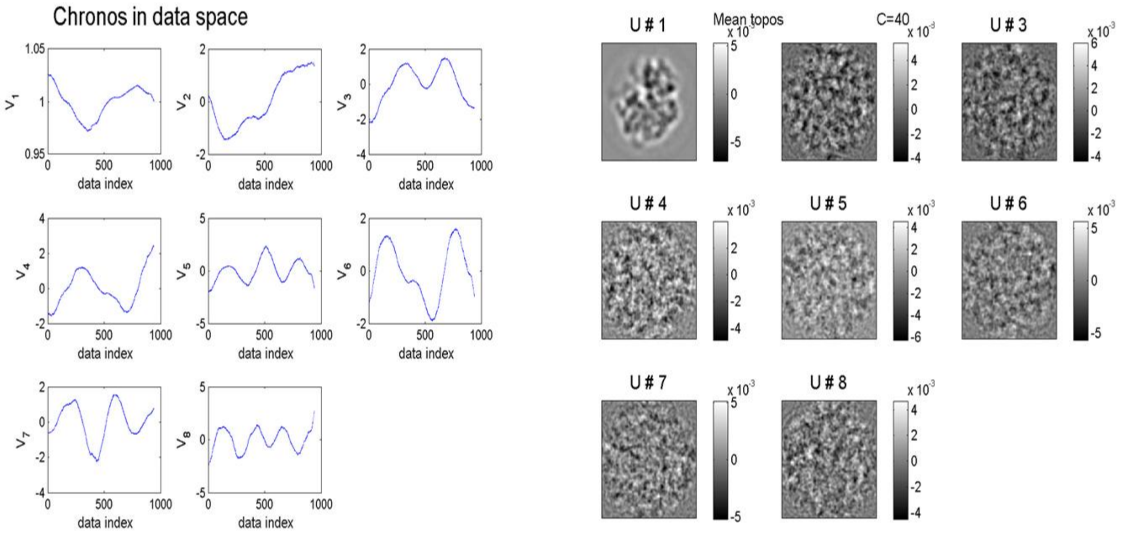


Figure 50: Topos (right) and chronos modes (left) of NLSA. They provide spatio-temporal information regarding evolution of an ordered one dimensional manifold using NLSA.

4.3.6. Connecting projection directions

Having a correct interpretation from the results of local embeddings relies upon their association with each other. For example, if local embedding of projection directions reveals two discrete clustering for each directions, members of each cluster must be associated along projection directions to entail sufficient information for further analysis of data and ultimately three-dimensional reconstruction.

The association between projection directions can be explained better by contour line analogy. The resulting one-dimensional manifold can be viewed as a iso-orientational contour line over each projection direction. By ruling out orientation signal and other systematic factors, manifold embedding of each projection direction reveals information about conformational changes of the molecule in that projection direction. However, in

www.manaraa.com

order to associate different manifolds, another contour line is required that contain the iso-conformational information along projection directions. This association is provided via finding the relation between metrics that describe manifolds in each projection directions. Finding this relationship is similar to finding the same coordinates for all manifolds of projection directions.

In the case of one-dimensional manifolds, the relationship between metrics can be found simply by considering the following principle: Having a one-dimensional manifold, the describing metric is proportional to the inverse density of points along the manifold direction. The relationship is found by equalizing the histograms of points densities along all one-dimensional manifolds found in projection directions.

Before finding the densities, manifold direction needs to be quantified. Eigenfunctions of an one-dimensional manifold can be best described by single harmonics in the following form:

$$\boldsymbol{\Psi_w = a_w \times \cos(w\pi\tau)}$$

First three dominant Eigenfunctions of embedded manifolds fit to single harmonics by an iterative nonlinear least square fit. Fitting starts with assigning random parameters (a_w and τ) to $\boldsymbol{\Psi_w}$ and continuing with nonlinear trust-region reflective least square fit. Then, every fitted point is projected into manifold and the projected distance is measured as the cost function for next iterations of nonlinear least square fit. By having the projected distance of fitting function we are giving more importance to the density difference between points along the one-dimensional manifold. Figure 51 shows the results of iterative fitting.

www.manaraa.com

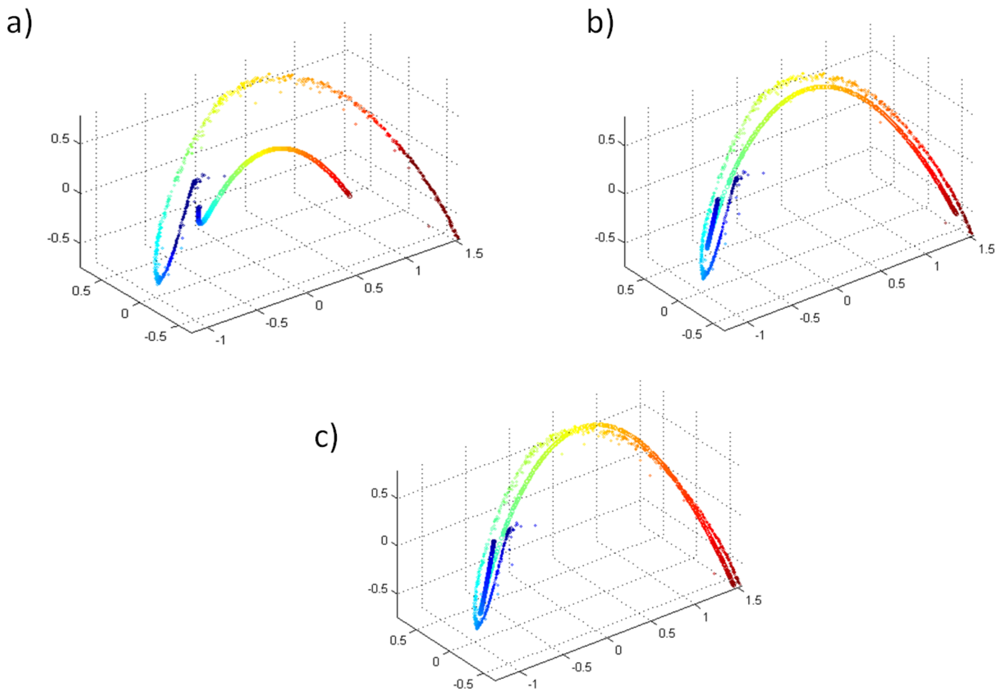


Figure 51: Different steps of the iterative fitting process. One-dimensional manifold (scattered) is fitted with single harmonics (Continuous). Manifold is ordered based on geodesic distances to lower left corner. a) First fitting step. b) Fitting after 8 iterations c) Best fit with lowest cost function after 20 iterations

As a result of fitting, every point in one-dimensional manifold has an associated τ-value which is a measure of its position in the curve. Figure 52 demonstrates the distribution of points along manifold direction for fifty projection directions.

المنارة للاستشارات

www.manaraa.com

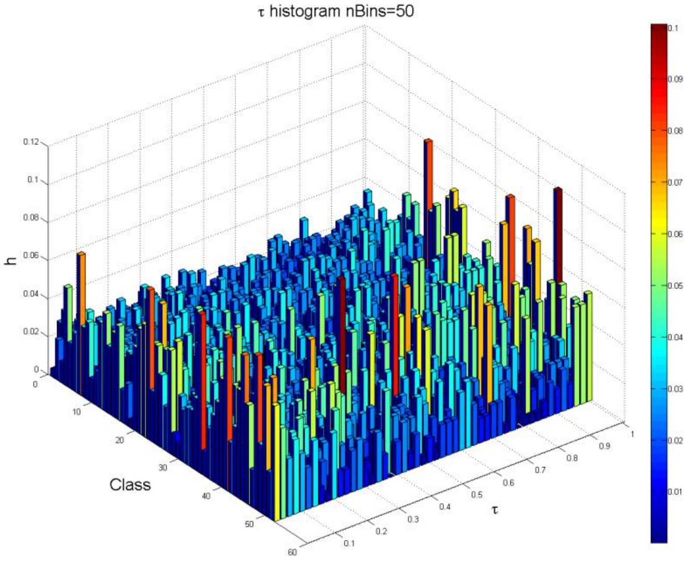


Figure 52: Histogram of τ distribution over projection directions. Points are color-coded based on their population in each histogram bin.

After finding a measure of moving along the manifold direction in each projection direction, next step is to connect different local measures to a global measure of one-dimensional change by using Histogram equalization. Histogram equalization assumes that two histograms are probability density functions (pdf) of two random variables of the same distribution and will enhance their pdfs towards a mean pdf. By equalizing histograms, the new τ values are the global measure of direction along manifold for all projection directions.

www.manaraa.com

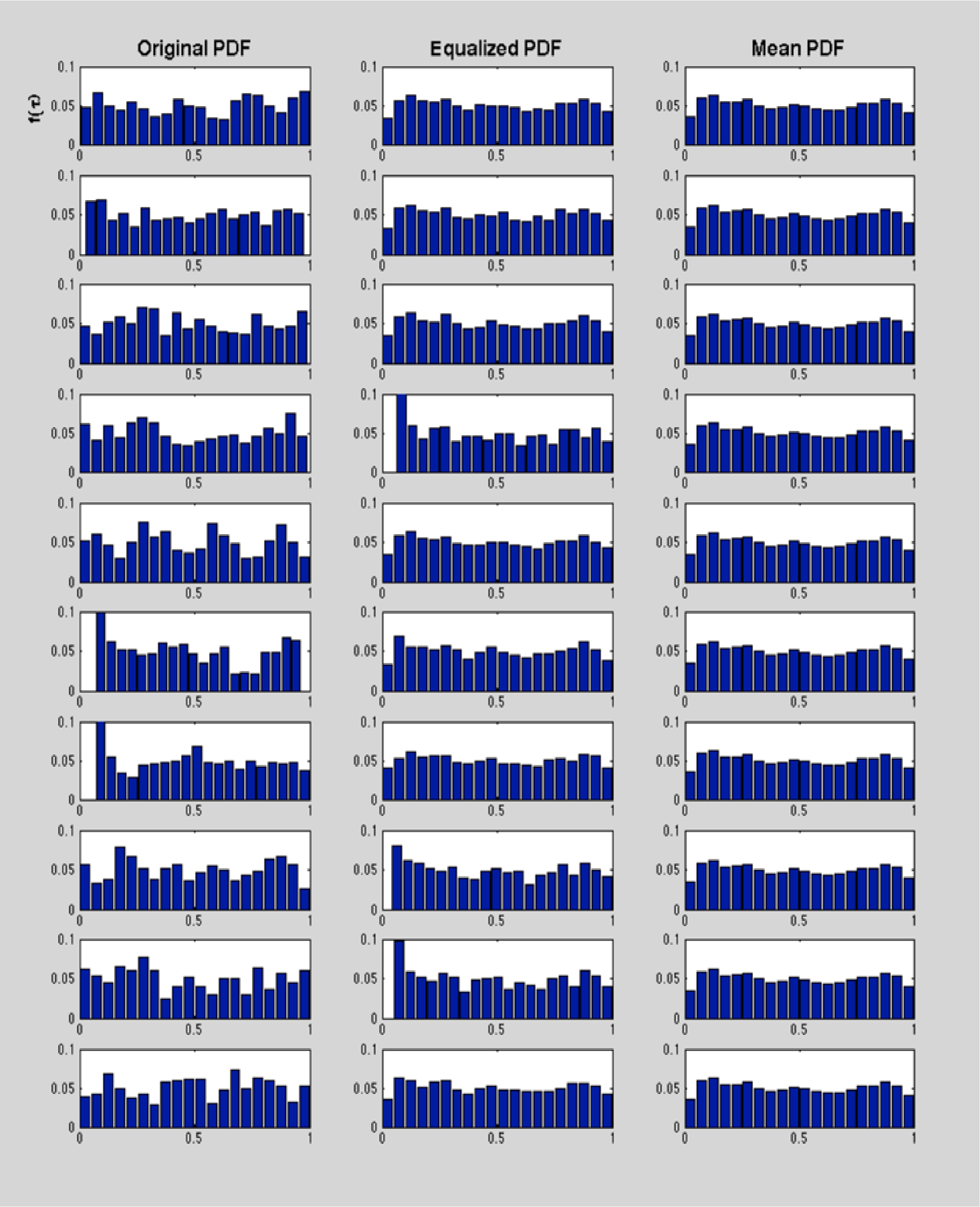


Figure 53: Results of Histogram equalization. Original distribution, Equalized τ histograms and mean distribution of first ten most populated projection direction with one-dimensional manifold are shown in the figure.

www.manaraa.com

4.4. Results

4.4.1. Local embeddings

Local embedding is performed on 189 projection direction with more than 1000 snapshots. As a results of embedding, three categories of manifolds were observed each containing roughly one third of the projection directions.

Class 1: Fifty five projection directions showed a manifold that entails a dense core (~80% of snapshots) and an one-dimensional tail (~20% of snapshots). (Figure 54)

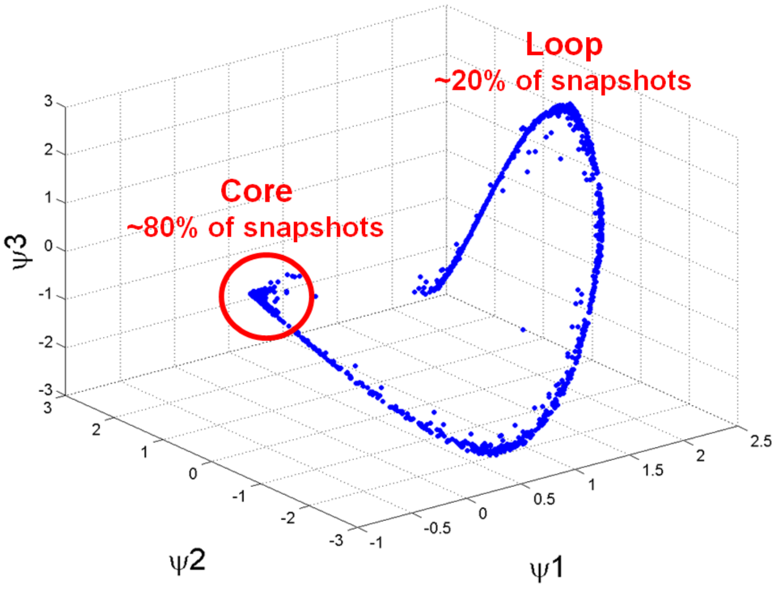


Figure 54: Local embedding of class 1. Scatter plot of projection snapshots with a one-dimensional tail (~20% of population) is demonstrated.

Class 2: Seventy three projection directions showed a manifold with a shapeless core and no one-dimensional tail (Figure 55).

www.manaraa.com

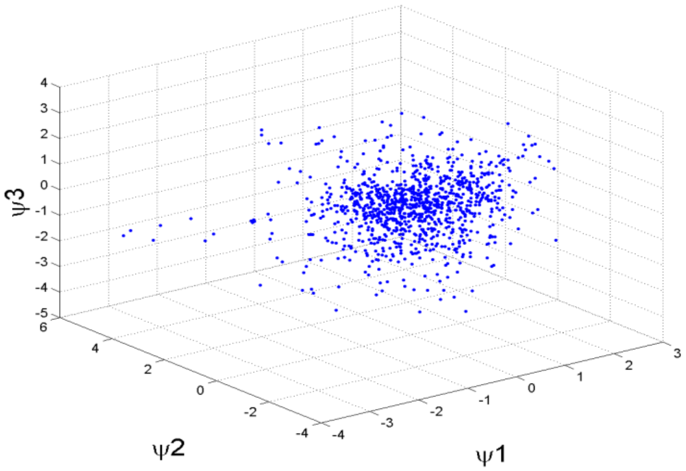


Figure 55: Local embedding of class 2. Embedding shows no manifold characteristics.

Class 3: Remaining sixty one projection direction presented a intermediate behavior with a dense core (~90%) and small shapeless tail (~10%) (Figure 56).

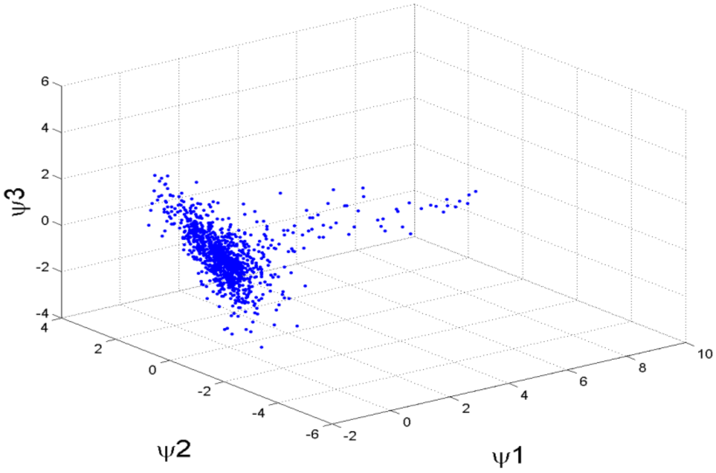


Figure 56: Local embedding of class 3. It represents manifolds with intermediate characteristics.

The distribution of three classes over projected S2 (two-dimensional) is shown in Figure 57:

www.manaraa.com

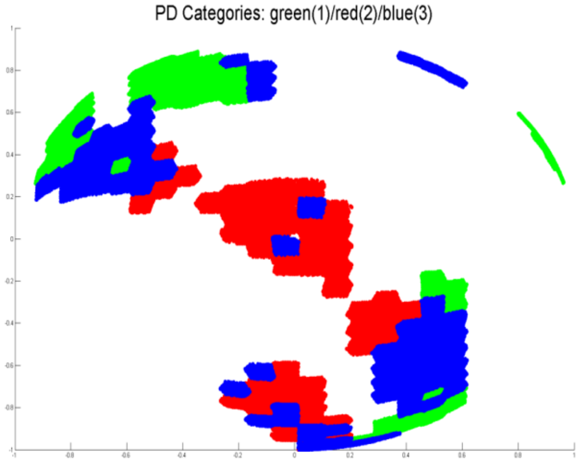


Figure 57: Distribution of three category of projection directions over S2. For better representation, top view projection of S2 is showed here. Green : Class 1, Red: Class 2 and Blue: Class 3

For further analysis, we focused our attention on class 1 projection directions. At first, we are interested in embedding characteristics of each section of manifold (core and one-dimensional tail) separately. For this purpose, snapshots in core and tail sections of the manifolds are re-embed using Diffusion Map. Re-embedding of the tail section reveals an approximate parabola in two-dimensional space which is a characteristic of manifold produced by single harmonics (more details in methods section).

www.manaraa.com

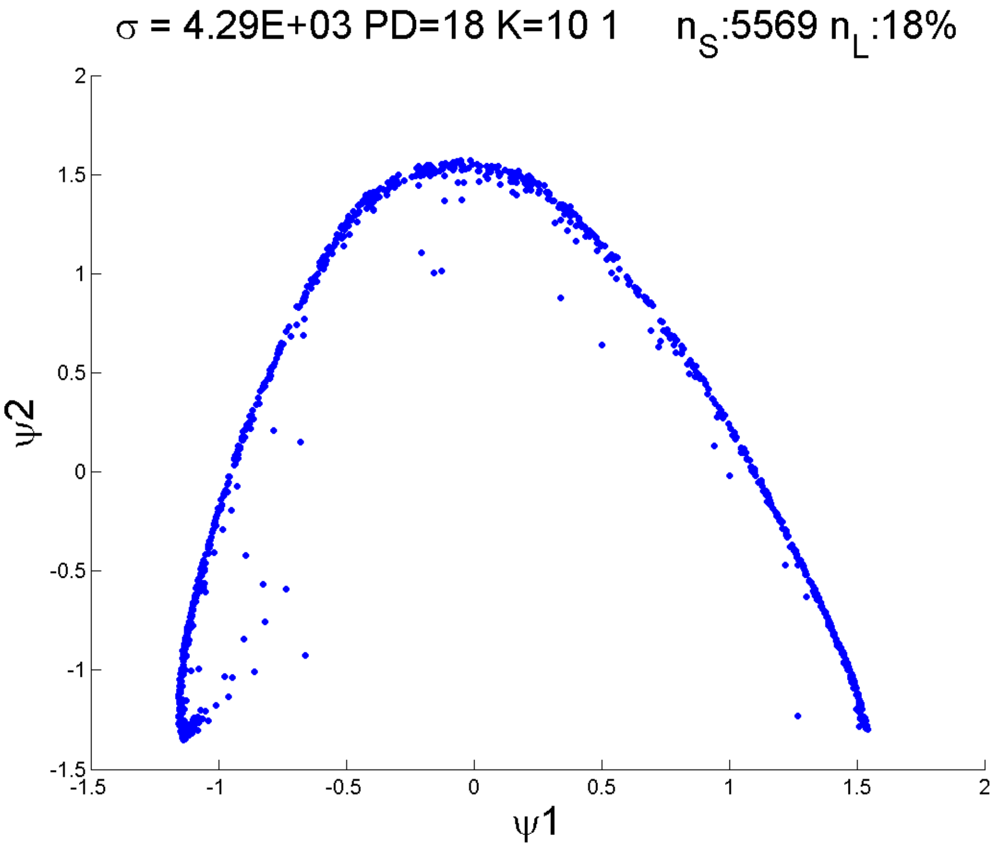


Figure 58: Re-embedding the tail manifold. Observed Parabola, best described by single harmonics, as a result of re-embedding one-dimensional tail of manifolds in class 1 projection directions. σ is the Gaussian width of embedding, PD: projection direction, k: number of nearest neighbors, ns: number of snapshots in projection direction and nl: Percentage of snapshots that belong to one-dimensional tail of the manifold.

Re-embedding of the core section reveals a two-dimensional sheet which by qualitative analysis reveals the information about the small orientation difference of the snapshots inside the projection direction.

المنارة للاستشارات

www.manaraa.com

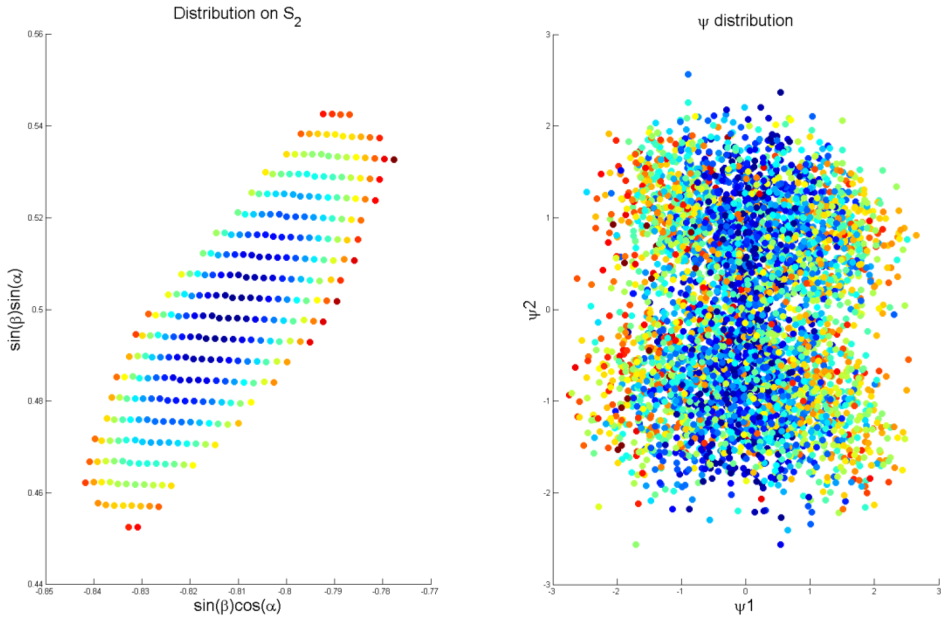


Figure 59: Re-embedding the core. Distribution of snapshots in S2 (left) and re-embedding the core section of manifold produced from class 1 projection direction (Right). As the figure shows, the re-embed manifold has the information regarding the orientation of snapshots. Point are color-coded based on their proximity to the center of left diagram.

.

4.4.2. Overrule systematic factors

Focusing on class 1 projection directions, we are interested in finding the origin of one-dimensional tail. We tried to find a correlation between known systematic factors and observed one-dimensional manifold but the correlation coefficients were insignificant ($r<0.15$) leading us to the conclusion that the one-dimensional manifold indeed represent conformational changes of the ribosome.

Pearson linear correlation coefficient is used for the measure of dependency between the geodesic distance of the points (representing the loop order) and known systematic measures.

At first defocus value of each snapshot and difference defocus between two snapshots are thought to cause the observed behavior. However correlation coefficients between geodesic distance over the manifold and defocus (Figure 60) and difference defocus (Figure 61) over the manifold were very small, ruling out the defocus as a causing factor.

Secondly for testing the validity of the assumption that one-dimensional manifold have a specific orientation pattern over the S2, snapshots distribution over S2 is plotted and no specific pattern is revealed (Figure 62).

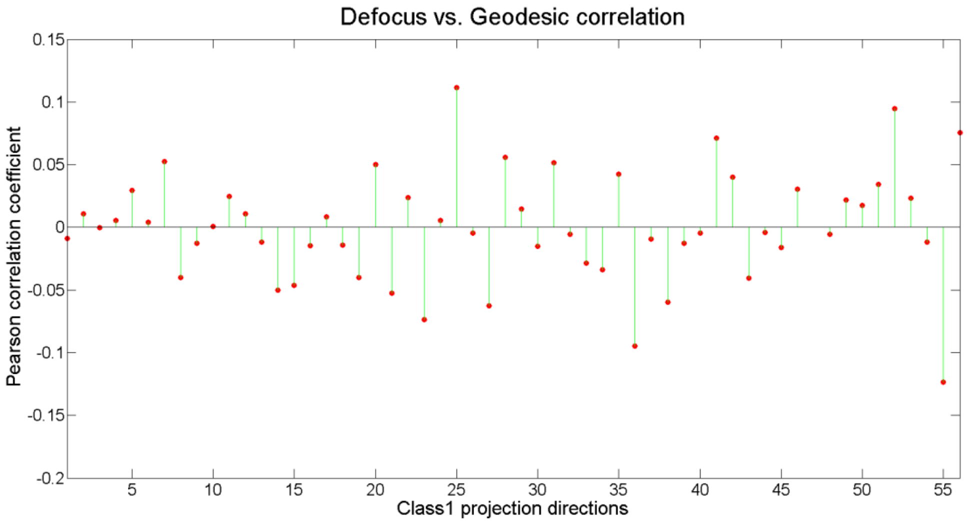


Figure 60: Correlation between direction order and defocus value. Note the insignificancy of the results |r|<0.12

www.manaraa.com

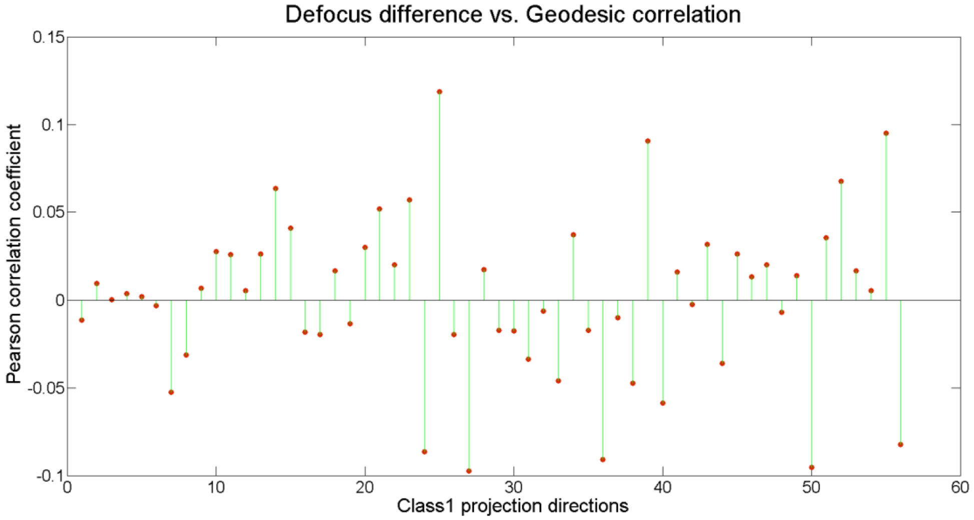


Figure 61: Correlation between direction order and difference defocus value. Note the insignificancy of the results |r|<0.13

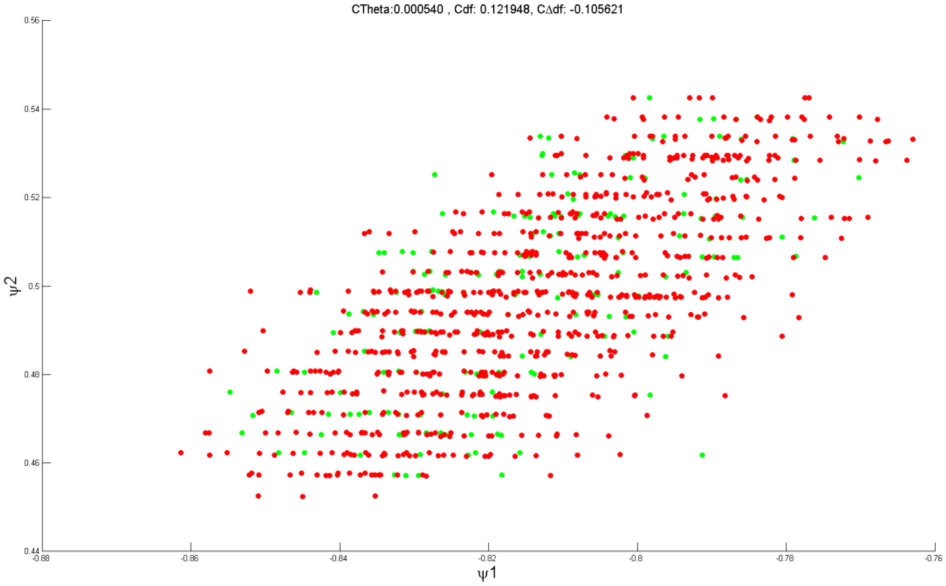


Figure 62: distribution of snapshots from tail of class 1 manifolds over S2. Green: one-dimensional snapshots (~20%), Red: Other snapshots (~80%). No significant pattern is produces there.

www.manaraa.com

Quantitatively, we measured the correlation between the angular distance between the points on S2 and geodesic distance over the direction of manifold and couldn't find a large correlation in any projection directions (Figure 63).

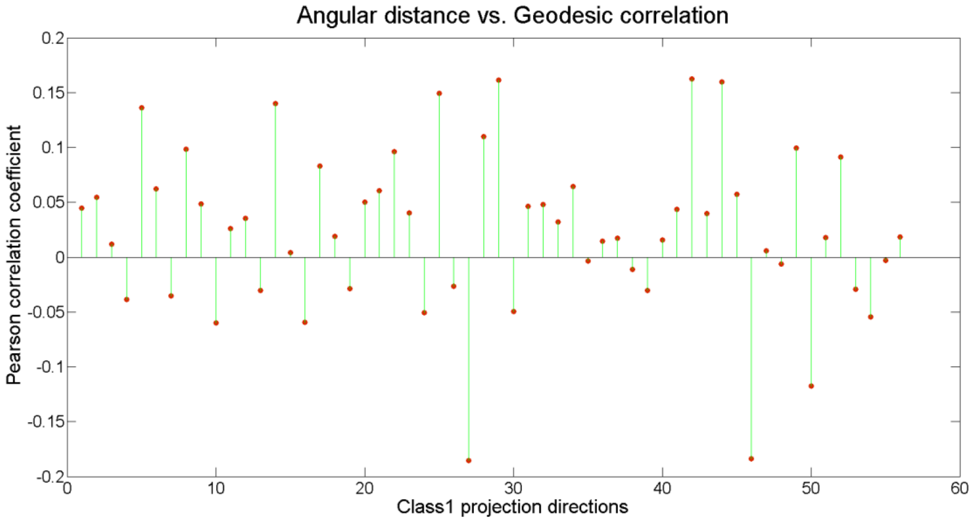


Figure 63: Correlation between direction order and angular distance over S2. Note the insignificancy of the results |r|<0.17

In another view of the relation between one-dimensional manifold and orientations over S2, Euclidean distance of each points from the projection direction center is used as a measure of orientation distance and qualitatively shown over the loop direction. Similar to re-embedding the core, as Figure 64 shows, having a relations between manifold and orientations cause the same pattern between snapshots over S2 and in embedded state which was not the case with one-dimensional manifolds.

المنارة للاستشارات

www.manaraa.com

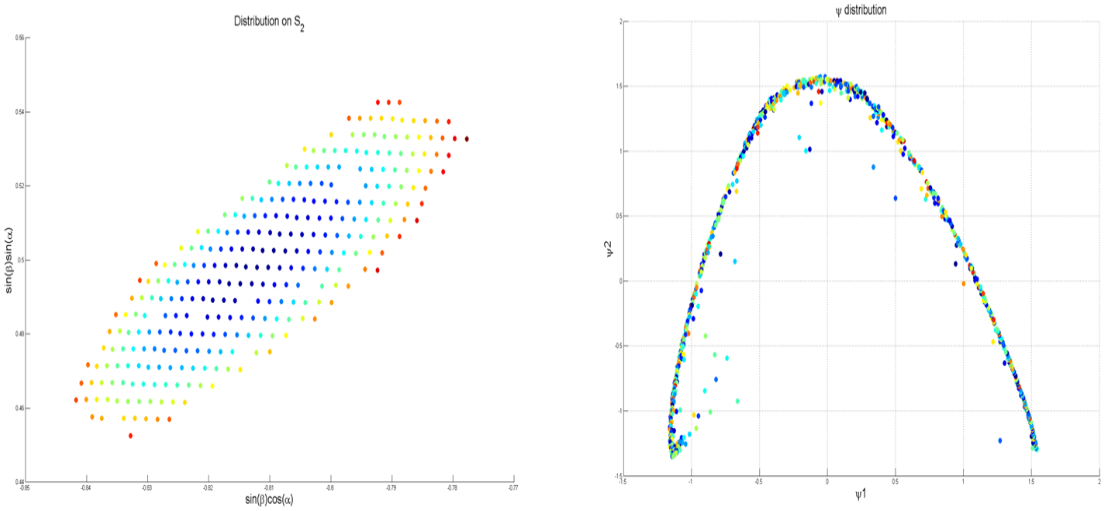


Figure 64: Distribution pattern of tail manifold. Distribution of snapshots in S2 (left) and re-embedding the tail section of manifold produced from class 1 projection direction (Right). As the figure shows, the re-embed manifold has creates no specific patterns with the orientation of snapshots. Point are color-coded based on their proximity to the center of left diagram.

In addition, observing the images by eye did not reveal any pathological behavior that cause separation between core and tail sections (Supplementary movie I[6]). Projection directions which entail class 1 manifold also didn't have any characteristic population pattern. They are distributed in some local hot spots over S2 might be indicative of larger conformational change that are observable in those directions.

Overall, we could not find a systematic factor that generates the tail section of class 1 manifold. This led us to this belief that the tail section is representative of a continuous conformational change in ribosome. From biological point of view, this notion can be supported by taking into account the continuous ratchet like motion of small subunit and

---

[6] http://goo.gl/wts2Nm

www.manaraa.com

its coupling motions of large subunit parts in empty ribosome (without t-RNA) and due to thermal fluctuations of energy. Capturing empty ribosome was one of the subjects of experimental study that led to cryo-EM dataset and further support this idea.

4.4.3. Iso-orienational movie

Focusing on projection directions with one-dimensional manifold (class1), we want to observe statistically significant differences between core and tail sections. Average images of the snapshots in core and tail sections does not reveal any specific differences (Figure 65). In addition, moving average of snapshots by walking in tail direction does not reveal any continuous observable changes (supplementary movie II[7]).

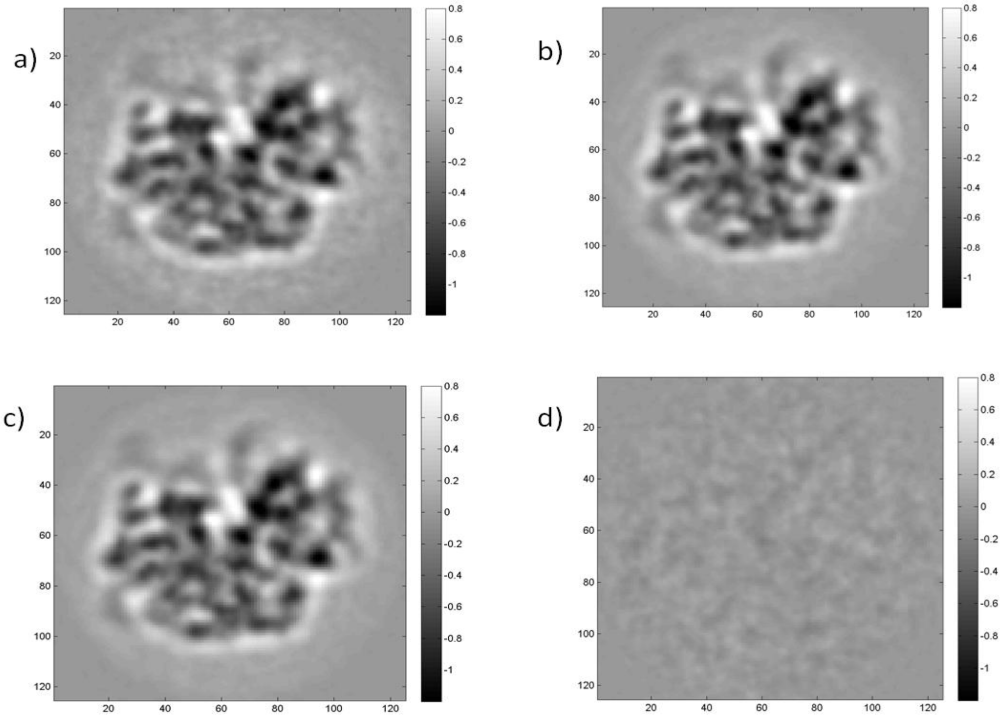


Figure 65: Average from tail (a), core (b) and all (c) snapshots. As the figure shows, averaging does not provide a significant difference between core and tail sections of manifold. d) demonstrate the intensity difference of tail and core averages.

---

[7] http://goo.gl/cHouml

المنارة للاستشارات
www.manaraa.com

However in order to capture the true nonlinear spatiotemporal dynamics of the manifold, we used NLSA. Using power shifting routine to enhance the power of first mode to become equal to sum of the power of all other modes, reconstructed snapshots of first eight modes of NLSA revealed concerted continuous change globally distributed over the whole image areas (supplementary movie III[8]).

The observed smooth change can also be observed from the behavior of the back-projected chronos modes that represent the temporal dynamics of snapshots manifold (Figure 66).

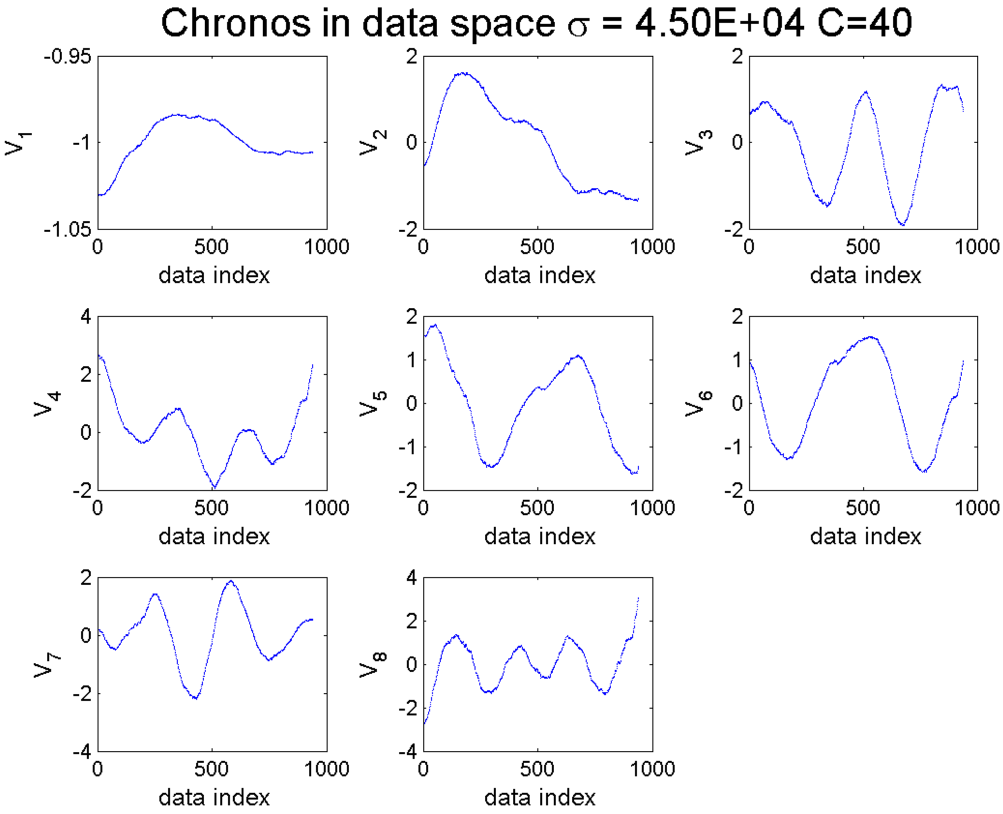


Figure 66: NLSA Chronos modes. It reveals the smooth, continuous change associated with walking through direction of one-dimensional manifold.

---

[8] http://goo.gl/HxqXU9

www.manaraa.com

4.4.4. Iso-conformational map

Iso-conformational map is the contour line that connect snapshots with same conformation over projection directions. Using iterative nonlinear least square fitting and histogram equalization, global τ values are extracted as a measure of movement along conformational direction (presumed direction of one-dimensional manifold) for all 55 projection directions of class 1 manifolds. In order to specify iso-conformational contours, two tasks must be done. First, a path need to be specified between projection directions. We chose a path closest to a great circle in order to enhance the tomographic three-dimensional reconstruction of conformational states (supplementary movie IV[9]). Secondly, separate conformational states need to be extracted for each contour line. By binning τ values into 10 bins, all snapshots are categorized into ten classes each representing a conformational state of ribosome and hence ten iso-conformational contour lines are drown. Supplementary movie V[10] demonstrates one of the iso-conformational paths.

---

[9] http://goo.gl/QiEHa2
[10] http://goo.gl/sEZXCN

المنارة للاستشارات
www.manaraa.com

4.5. Conclusions

This chapter presents a theoretical framework and algorithmic tool-chain for capturing the dynamics of ribosome from cryo-EM experimental data. Applying Diffusion Map embedding to local projection directions (small tessellations of S2 orientation space), gives a one-dimensional manifold which to our beliefs represents the continuous conformational changes of empty ribosome due to thermal fluctuations. By using NLSA, we were able to capture the temporal dynamics of this manifold. The concerted, smooth dynamics of this manifold is presented as a iso-orientation movie of conformational changes over 2D snapshot of each projection direction. In addition, we proposed a framework for connecting one-dimensional manifolds of different projection directions over S2. A global measure of continuous conformational change is used for each snapshot as a result of connection. Having this parameter, we were able to draw iso-conformational contours over projection directions over S2 connecting all the snapshots with the same conformation state and different orientations. Iso-conformational contour will help the future three-dimensional reconstruction of each conformational state. As a result of this work, we developed a pioneering approach to capture the structural variability of ribosome and its continuous functional states in a completely unsupervised manner.

المنارة للاستشارات

www.manaraa.com

4.6. References

Christiansen, Bo. "The shortcomings of nonlinear principal component analysis in identifying circulation regimes." *Journal of climate* 18.22 (2005): 4814-4823.

Dee, D. P., et al. "The ERA-Interim reanalysis: Configuration and performance of the data assimilation system." *Quarterly Journal of the Royal Meteorological Society* 137.656 (2011): 553-597.

Fischer, Niels, et al. "Ribosome dynamics and tRNA movement by time-resolved electron cryomicroscopy." *Nature* 466.7304 (2010): 329-333.

Giannakis, Dimitrios, and Andrew J. Majda. "Nonlinear Laplacian spectral analysis for time series with intermittency and low-frequency variability." *Proceedings of the National Academy of Sciences* 109.7 (2012): 2222-2227.

Giannakis, Dimitrios, and Andrew J. Majda. "Nonlinear Laplacian spectral analysis: capturing intermittent and low-frequency spatiotemporal patterns in high-dimensional data." *Statistical Analysis and Data Mining* (2012).

Hsieh, William W. "Nonlinear principal component analysis of noisy data." *Neural Networks* 20.4 (2007): 434-443.

Lima, Carlos HR, et al. "Statistical prediction of ENSO from subsurface sea temperature using a nonlinear dimensionality reduction." *Journal of Climate* 22.17 (2009): 4501-4519.

Lovisolo, L., and E. A. B. Da Silva. "Uniform distribution of points on a hyper-sphere with applications to vector bit-plane encoding." *IEE Proceedings-Vision, Image and Signal Processing* 148.3 (2001): 187-193.

Shaikh, Tanvir R., et al. "Particle-verification for single-particle, reference-based reconstruction using multivariate data analysis and classification." *Journal of structural biology* 164.1 (2008): 41-48.

Tenenbaum, Joshua B., Vin De Silva, and John C. Langford. "A global geometric framework for nonlinear dimensionality reduction." *Science* 290.5500 (2000): 2319-2323.

Vautard, Robert, and Michael Ghil. "Singular spectrum analysis in nonlinear dynamics, with applications to paleoclimatic time series." *Physica D: Nonlinear Phenomena* 35.3 (1989): 395-424.

المنارة للاستشارات

www.manaraa.com

Chapter 5

Conclusions and future directions

www.manaraa.com

5. Conclusions and future direction

5.1. Conclusions

In this thesis, a theoretically sound and computationally powerful algorithmic chain is proposed to tackle the problem of high-resolution structure recovery in the presence of structural heterogeneity from single particle imaging techniques such as Cryogenic Electron Microscopy (cryo-EM) and emerging X-ray Free Electron Laser (XFEL).

A review on existing algorithmic tools for recovering high resolution structure revealed their limitations for reconstruction of large macromolecules and viruses due to their i) theoretical and ii) computational constraints. In addition and in the presence of heterogeneity, there is a need for an algorithmic tool capable of iii) unsupervised detection of continuous conformational changes of complex molecular machines such as ribosome. All these issues are addressed in the course of this thesis.

By the advent of XFEL, there is a constant increase in flow of data as the number of diffraction snapshots per equipment run is increasing. Hereby an expandable and reliable set of algorithms to overcome the computational expenses of data processing becomes a necessity.

In chapter two, a fast and extendable implementation of Diffusion Map embedding is proposed using computational power of Graphical Processing Units (GPU). The developed algorithm forms a data and compute intensive software pipeline for structure

المنارة للاستشارات
www.manaraa.com

and conformation recovery of biological entities from a large datasets of Cryo-EM and XFEL.

The importance of developed pipeline is demonstrated in two main applications presented in Chapter three. a) the first approach capable of extracting high-resolution 3D structure from diffraction snapshots of symmetric objects with structure recovery to 1/100th of the object diameter and b) the hitherto impossible analysis of 20 million simulated diffraction snapshots for capturing conformational heterogeneity of Adenylate kinase (ADK) molecule in the process of molecule melting.

we have outlined powerful new algorithmic approaches based on Diffusion Map embedding, and demonstrated their potential for extracting signal from noise, recovering 3D structure with no orientation information, separating discrete conformations and species, and eventually mapping conformational continua.

Final chapter of thesis presents a solution for the latter challenge by extracting and processing continuous structure v ariablity of ribosome from experimental cryo-EM data in the presence of overwhelming noise(SNR≈ -12 DB).

5.2. Discussions

As far as the algorithm development goes, there is room for further enhancements in our implementation, as evidenced from comparing the raw float point processing power of the processors. The most expensive part of the brute force *k*-NNG is matrix multiplication. With the best tuned GPU libraries, we see that there is only 50 percent use of GPU resources as opposed to 90 percent use by finely tuned CPU libraries. A better

www.manaraa.com

GPU matrix multiplication library would further enhance the performance of our approach.

The brute force implementation requires $O(n^2)$ distance calculations. This dominates the computational expense for a large n. One way of reducing this complexity will be to use a hybrid algorithm that approximately subdivides the data sets into overlapping sets to reduce the computations (distance computation and selection) for each input vector based on the set membership. While such methods exist, to our knowledge there are no parallel cluster implementations with GPU acceleration.

Finally, performance of the algorithms is significantly impacted by the nature of the data (data dimension as well as size) and the execution configuration parameters. For example, currently we manually select the number of data partitions P of the input matrix A and the size of sub-partitions of AI within the nodes. The GPU quick select algorithm in particular is very sensitive to the number of simultaneous queries and the arrangement of execution resources (the number of warps per thread block). Currently, these execution parameters are manually set and adjusted through trial and error. This can be automated is the future.

High-resolution structure recovery in the presence of symmetry has proven to be a powerful tool for recovering structure by enhancing the effective number of snapshots and improving resolution. In addition by having a superior noise robustness, our approach for determining high-resolution structure at low radiation signal levels will play a vital role in the future of structure recovery from single particle techniques and understanding "biologically relevant" information from biological entities.

المنارة للاستشارات

www.manaraa.com

However, preliminary results of our analysis of XFEL diffraction snapshots does not reveal meaningful structural information. The main reason for this issue is the presence of experimental artifacts, such as variations in beam intensity, position and inclination, and other systematic nonlinearities and limitations. Nevertheless by generating more datasets of biological entities and the rapid progress in XFEL-based imaging and processing techniques, improved single-particle XFEL datasets are expected in near future.

One immediate application of our manifold-based approach for mapping discrete conformations is the purification of different biological molecules or even molecular-species after the experiment. Using this algorithm can potentially lead to simpler experimental sample preparation and faster experimental setups for generating snapshots of single particles. In addition by providing a processing pipeline for 20 million snapshots, our approach is showing superior capabilities for analyzing current state of the art datasets of biological entities and set the stage for upcoming results of future XFEL experiments.

For studying continuous conformational changes, our manifold-based approach is perhaps most appropriate algorithm, because it requires no *a priori* knowledge of the type and characteristics of hidden conformational information and has the capability to deduce conformational information by finding the governing dimensions of 2D snapshots in a totally unsupervised manner. This is in contrast to Bayesian approaches that either require prior knowledge of the number of conformations, or deduce this number by trial and error.

www.manaraa.com

5.3. Future directions

Manifold-based dimensionality reduction approaches and specially Diffusion Map has been providing theoretically sound, noise robust and expandable solutions for processing the output of single particle imaging modalities such as XFEL and CryoEm. In future, these algorithms has the potential for inspiring structural studies of even more complex systems such as Mitochondria, Endoplasmic reticulum, Chromosomes and even the whole cell.

The capability of manifold-based approaches to map conformation continua from single particle snapshots will have a revolutionizing effect in both methodology and biological understanding of molecular machines and their interactions in cellular environment. Upon their success, these methods has the potential for building the new understanding of dynamics in molecular and cellular level and shape the future of cellular physiology and molecular biology. This thesis has taken a small but important step towards this goal by providing computational boost, initial proof of concept in simulation environment and parameterizing an observed continuous conformational phenomena in ribosome Cryo-EM data.

In this regards, our immediate future step is to have a biologically sound interpretation of the observed behavior by having a 3D reconstruction of conformational parameterized intermediate steps and matching the observed behavior to knowledge base of previously found functional states of ribosome. For this purpose, a reconstruction method capable of 3D reconstruction of conformational classes in the presence of severe orientation anisotropy and sparsity of conformational information in orientation space is a necessity.

المنارة للاستشارات
www.manaraa.com

In addition, there are room for improvement in other aspects of this work as well. In algorithm development, we achieved the computational capability to deal with data ensembles at least one order of magnitude larger than current datasets. However, with the growing pace of expansion in input data sizes (both in number of dimensions and reference data points) alongside technological advances in GPU hardware and software configurations, new strategies for data management, partitioning and algorithm development will be needed in the foreseeable future. Our future algorithm development will continues in two different areas. First is the development of a stand-alone *k*-NN search library on GPU that outperforms all state of the art algorithms and that can be easily accessible and applicable to huge applications of *k*-NN search and *k*- NN graph construction. In order to achieve this goal, we are planning to create a user friendly environment and a parameter space for different range of applications. The second possible target of future work in this area is to release a user friendly version of a *k*-NNG construction library for GPU distributed systems. The proposed library would use the expandability and flexibility of proposed algorithm with regards to computational system configurations and data characteristics respectively.

We also provided a strong footing for manifold-based approaches in the future of XFEL data analysis. Our both model systems for structure recovery and mapping conformational changes, in addition to our computationally powerful implementation, set the stage for analyzing future XFEL high throughput diffraction snapshots. One future avenue of this thesis would be to apply the proposed methodology for high-resolution structure recovery of symmetric viruses and mapping conformational states of large

المنارة للاستشارات
www.manaraa.com

proteins and macromolecules from XFEL experiments, upon the quality improvement and resolving the systematic artifacts.

المنارة للاستشارات
www.manaraa.com

6. Curriculum vitae

Ali Dashti

Education:

University of Wisconsin - Milwaukee, PhD in Biomedical and Healthcare Informatics, Total GPA: 3.87/4.00

University of Wisconsin - Milwaukee, M.S. in Electrical Engineering

Sharif University of Technology, Tehran, Iran, M.S. in Biomedical Engineering, Total GPA: 16.68/20

Sharif University of Technology, Tehran, Iran, B.S in Electrical Engineering, Total GPA: 15.61/20

Academic Experiences:

PhD dissertation: Extracting the structure and conformations of biological entities from large datasets

Electrical Engineering Master's thesis: Efficient Computation of k-Nearest Neighbor Graphs for Large High-dimensional Data Sets on GPU Clusters.

Biomedical Engineering. Master's thesis: Investigating the effect of electroporation of living tissue on ultrasound signals.

Electrical Engineering Bachelor's thesis: Classification of abstract thinking tasks by processing EEG signals.

Publications:

PLoSOne (Submitted): I. Komarov, A. Dashti, R. M. D’Souza, "Fast k-NNG construction with GPU-based quick multi-select."

Philosophical transaction in biological sciences (Under revision): A. Hosseinizadeh, P. Schwander, A. Dashti, R. Fung, R.M. D’Souza, A. Ourmazd, "High-Resolution Structure of Viruses from Random Snapshots."

PLoSOne: A. Dashti, I. Komarov, R. M. D'Souza, "Efficient Computation of k-Nearest Neighbor Graphs for Large High-dimensional Data Sets on GPU Clusters."

Proceeding of CHS research symposium, UWM: A. Dashti, C. Murphy, R. M. D’Souza, E. Gafni, , Z. Cai, , P. J. Tonellato, "Graphical Processing Units, Next Generation Sequencing, and the Future of Health Care Informatics."

Journal of Computer Science and Systems Biology: A. Dashti, R.Ranjbar, M. Hajihasani, S. Gharibzadeh. "Design of a Gene Tuning motif inspired by stem cell core rheostat."

المنارة للاستشارات

www.manaraa.com